



TUGAS AKHIR - TE 141599

**RANCANG BANGUN PERANGKAT LUNAK UNTUK
MENGUBAH MODEL 3 DIMENSI MENJADI G-CODE
BAGI MESIN CNC**

Stefan Feliciano Tandawidjaja
NRP 2210100143

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Rudy Dikairono, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 141599

**SOFTWARE DEVELOPMENT FOR CONVERTING
3-D MODEL TO G-CODE FOR CNC MACHINE**

Stefan Feliciano Tandawidjaja
NRP 2210100143

Advisor
Dr. Ir. Djoko Purwanto, M.Eng.
Rudy Dikairono, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2015

**RANCANG BANGUN PERANGKAT LUNAK UNTUK
MENGUBAH MODEL 3 DIMENSI MENJADI G-CODE
BAGI MESIN CNC**

TUGAS AKHIR

**Diajukan Untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I



Dr. Ir. Djoko Purwanto, M.Eng.

NIP. 19810105200501104

Dosen Pembimbing II



Rudy Dikairono, ST., MT.

NIP. 196204181990031004



RANCANG BANGUN PERANGKAT LUNAK UNTUK MENGUBAH MODEL 3 DIMENSI MENJADI G-CODE BAGI MESIN CNC

Nama : Stefan Feliciano Tandawidjaja
Pembimbing I : Dr. Ir. Djoko Purwanto, M.Eng.
Pembimbing II : Rudy Dikairono, ST., MT.

ABSTRAK

Mesin CNC adalah mesin yang pergerakan dan fungsi dari *tool*–*tool*nya dikontrol melalui sebuah program yang berisi kode data alfanumerik. Teknologi seperti CNC dapat dimanfaatkan bidang industri khususnya industri kreatif yang identik dengan benda atau produk yang berbentuk unik dan kompleks dalam jumlah yang banyak. Mesin CNC sangat dibutuhkan karena dewasa ini, kebutuhan dan permintaan akan benda atau produk yang memiliki nilai seni semakin meningkat. Produk-produk seni sangat dibutuhkan pada berbagai bidang, khususnya pada bidang disain interior yang menghasilkan produk, yang biasanya berupa furnitur, dengan motif dan bentuk yang unik.

Perangkat lunak dirancang menggunakan *integrated development environment* atau IDE berupa Lazarus untuk membaca data berupa *vertex* dan *face* dari model 3 dimensi yang dibangun pada perangkat lunak animasi Blender. Data yang didapat, kemudian akan diolah menjadi titik-titik koordinat kartesian yang berfungsi sebagai jalur pergerakan *tool* dari mesin CNC. Dengan koordinat kartesian yang didapat, G-code yang sesuai akan dihasilkan untuk disimulasikan pada perangkat lunak untuk simulasi mesin CNC, yaitu OpenSCAM, sebelum diberikan pada mesin CNC.

Pengujian dilakukan pada 8 buah data yang merupakan grafis komputer 3 dimensi yang dikelompokkan pada tiga kategori, yaitu model sederhana, model dengan tingkat kerumitan sedang, dan model kompleks. Waktu proses mesin CNC rata-rata ketiga kategori pada material berbahan *medium density fibreboard* atau MDF dengan luas 30 cm x 30 cm adalah 535 detik, 1361 detik, dan 3236 detik.

Kata Kunci : CNC, Perangkat Lunak, Model 3-D.

-----Halaman ini sengaja dikosongkan-----

SOFTWARE DEVELOPMENT FOR CONVERTING 3-D MODEL TO G-CODE FOR CNC MACHINE

Name : Stefan Feliciano Tandawidjaja
1st Advisor : Dr. Ir. Djoko Purwanto, M.Eng.
2nd Advisor : Rudy Dikairono, ST., MT.

ABSTRACT

A CNC machine is a machine which tools' movements and functions are controlled by a program that contains alphanumeric data code. Technology like the CNC can be used on many industrial field, especially in the creative industry known for its unique and complex product in a large amount. CNC machine are needed nowadays because the demand for products with artistic values are increasing. Products with artistic values are needed on many fields, especially interior design, which produce product, usually furnitures, with unique form and motif. To simplify the process of cutting or forming the motif and form, a software is designed in this thesis.

Software are designed using integrated development environment or IDE, Lazarus, to read vertex and face data from 3-D models constructed from animation software, Blender. The acquired data are then processed and converted to Cartesian coordinates which function as a toolpath for CNC machine. With the acquired Cartesian coordinates, the G-code will be simulated on a CNC simulator, OpenSCAM, before fed to the CNC Machine.

The experiments are conducted on eight piece of data which are 3-D computer graphics that are grouped into three categories, simple model, medium complexity model, and complex model. The average CNC machine process time for the three categories on a medium density fibreboard or MDF with 30 cm x 30 cm size are 535 seconds, 1361 seconds, and 3236 seconds.

Keywords : CNC, Software, 3-D Model.

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena berkat rahmat-Nya, penelitian dalam tugas akhir ini bisa berjalan lancar dan selesai tepat pada waktunya.

Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak, dan penulis sampaikan rasa terima kasih. Terima kasih yang sebesar-besarnya juga penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya :

1. Kedua orang tua tercinta, yang tidak pernah putus untuk seluruh do'a, nasihat, motivasi, dan dukungannya. Serta adik tercinta yang selalu menjadi sumber semangat.
2. Bapak Ronny Mardiyanto, ST., MT., PhD selaku dosen pembimbing pertama, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Bapak Dr. Ir Djoko Purwanto, M.Eng. selaku dosen pembimbing kedua, atas bimbingan, inspirasi, pengarahan, dan banyak pengalaman yang diberikan selama pengerjaan penelitian tugas akhir ini.
4. Rekan-rekan laboratorium b202 dan e-51 yang selalu memberi semangat dan keceriaan dan tiada henti membantu penulis dalam pada saat penelitian tugas akhir ini

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 22 Januari 2015

Stefan Feliciano Tandawidjaja

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL	
PERNYATAAN KEASLIAN TUGAS AKHIR	
LEMBAR PENGESAHAN	
ABSTRAK.....	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	ix
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan.....	2
1.5. Metodologi	2
1.6. Sistematika Penulisan	3
1.7. Relevansi	4
BAB 2 DASAR TEORI DAN TINJAUAN PUSTAKA	5
2.1. Model 3 Dimensi	5
2.1.1. <i>Vertex</i>	5
2.1.2. <i>Edge</i>	6
2.1.3. <i>Face</i>	7
2.1.4. Wavefront .obj File	8
2.2. <i>G-Code</i>	10
2.2.1. Kode.....	11
2.2.2. Variabel.....	12
2.2.3. Format	12
2.3. Perangkat Lunak ‘Blender’	13
2.4. Perangkat Lunak ‘OpenSCAM’	14
2.5. Perangkat Lunak ‘Lazarus’	15
2.6. Perangkat Lunak ‘Mastercam’	15
BAB 3 PERANCANGAN SISTEM	17
3.1. Pembacaan <i>File .obj</i>	18
3.2. Pengolahan Data	21
3.2.1. Pengolahan <i>Vertex</i>	21
3.2.2. Pengolahan <i>Face</i>	23
3.2.3. Pengolahan Akhir.....	25

3.3. Penulisan <i>File</i> bagi Mesin CNC	30
BAB 4 PENGUJIAN DATA	33
4.1. Perangkat Keras yang Digunakan	33
4.2. Model Sederhana	34
4.2.1. Data Pertama	34
4.2.2. Data Kedua	37
4.2.3. Data Ketiga	39
4.3. Model dengan Tingkat Kerumitan Sedang	42
4.3.1. Data Pertama	42
4.3.2. Data Kedua	45
4.3.3. Data Ketiga	48
4.4. Model Kompleks	51
4.4.1. Data Pertama	51
4.4.2. Data Kedua	53
4.5. Proses <i>Embossing</i>	56
4.5.1. Data Pertama	56
4.5.2. Data Kedua	58
4.6. Analisa Data	61
4.6.1. Kuantitas	61
4.6.2. Kualitas	62
BAB 5 PENUTUP	65
5.1. Kesimpulan	65
5.2. Saran	65
DAFTAR PUSTAKA	67
LAMPIRAN	69
BIODATA PENULIS	83

DAFTAR GAMBAR

Gambar 1.1 Produk industri kreatif.....	1
Gambar 1.2 Metodologi pengerjaan tugas akhir	2
Gambar 2.1 <i>Vertex</i> pada sebuah kubus	6
Gambar 2.2 <i>Edge</i> pada sebuah kubus.....	6
Gambar 2.3 <i>Face</i> pada sebuah kubus.....	7
Gambar 2.4 <i>Face</i> dengan lubang.	7
Gambar 2.5 Hasil dari <i>G-code</i> segitiga	13
Gambar 2.6 Logo dari Blender.....	14
Gambar 2.7 Logo dari OpenSCAM	14
Gambar 2.8 Lazarus	15
Gambar 3.1 Keseluruhan sistem	17
Gambar 3.2 Proses perangkat lunak.....	17
Gambar 3.3 Ilustrasi pembacaan file .obj.....	19
Gambar 3.4 Diagram alir proses pembacaan <i>file</i> .obj	20
Gambar 3.5 Ilustrasi pergeseran nilai menjadi positif.....	21
Gambar 3.6 Penskalaan model	22
Gambar 3.7 Pengolahan <i>vertex</i>	23
Gambar 3.8 Ilustrasi dalam mendapatkan <i>edge</i>	24
Gambar 3.9 Ilustrasi dua <i>edge</i> yang sama pada dua <i>face</i> yang berbeda	24
Gambar 3.10 Diagram alir proses pengolahan <i>face</i>	25
Gambar 3.11 Ilustrasi pemindahan <i>vertex</i> ke <i>array</i> akhir	26
Gambar 3.12 Ilustrasi penggunaan <i>array</i>	26
Gambar 3.13 Ilustrasi perubahan nilai <i>z</i>	27
Gambar 3.14 Ilustrasi proses penyederhanaan <i>final vertex</i>	28
Gambar 3.15 Diagram alir proses pengolahan akhir	30
Gambar 3.16 Penulisan <i>file</i> .gcode.....	31
Gambar 3.17 Diagram alir penulisan <i>file</i> .gcode.....	32
Gambar 4.1 Gambar 2 dimensi dari data pertama.....	35
Gambar 4.2 Model 3 dimensi dari data pertama	35
Gambar 4.3 Hasil simulasi dari data pertama.....	36
Gambar 4.4 Obyek yang dihasilkan mesin CNC	36
Gambar 4.5 Gambar 2 dimensi dari data kedua	37
Gambar 4.6 Model 3 dimensi dari data kedua.....	38
Gambar 4.7 Hasil simulasi dari data kedua.....	38
Gambar 4.8 Obyek yang dihasilkan mesin CNC	39
Gambar 4.9 Gambar 2 dimensi dari data ketiga.....	40

Gambar 4.10 Gambar 4.11 Model 3 dimensi dari data ketiga	40
Gambar 4.12 Hasil simulasi dari data ketiga	41
Gambar 4.13 Obyek yang dihasilkan mesin CNC	41
Gambar 4.14 Gambar 2 dimensi dari data pertama	43
Gambar 4.15 Model 3 dimensi dari data pertama	43
Gambar 4.16 Hasil simulasi dari data pertama	44
Gambar 4.17 Obyek yang dihasilkan mesin CNC	45
Gambar 4.18 Gambar 2 dimensi dari data kedua.....	46
Gambar 4.19 Gambar 4.20 Model 3 dimensi dari data kedua	47
Gambar 4.21 Hasil simulasi dari data kedua.....	47
Gambar 4.22 Obyek yang dihasilkan mesin CNC	48
Gambar 4.23 Gambar 2 dimensi dari data ketiga.....	49
Gambar 4.24 Gambar 4.25 Model 3 dimensi dari data ketiga	49
Gambar 4.26 Hasil simulasi dari data ketiga	50
Gambar 4.27 Obyek yang dihasilkan mesin CNC	50
Gambar 4.28 Gambar 2 dimensi dari data pertama	51
Gambar 4.29 Model 3 dimensi dari data pertama	52
Gambar 4.30 Hasil simulasi dari data pertama	52
Gambar 4.31 Obyek yang dihasilkan mesin CNC	53
Gambar 4.32 Gambar 2 dimensi dari data kedua.....	54
Gambar 4.33 Gambar 4.34 Model 3 dimensi dari data kedua	54
Gambar 4.35 Hasil simulasi dari data kedua.....	55
Gambar 4.36 Obyek yang dihasilkan mesin CNC	55

DAFTAR TABEL

Tabel 4.1 Spesifikasi Excitech SHG 1224 ^[9]	33
Tabel 4.2 <i>Face</i> , <i>vertex</i> , dan waktu proses dari ketiga kategori model...	61

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dewasa ini, kebutuhan dan permintaan akan benda atau produk yang memiliki nilai seni semakin meningkat. Produk-produk seni sangat dibutuhkan pada berbagai bidang, khususnya pada bidang disain interior yang menghasilkan produk, yang biasanya berupa furnitur, dengan motif dan bentuk yang unik seperti pada gambar 1.1. Karena semakin meningkatnya permintaan tersebut, kebutuhan akan mesin CNC untuk produksi massal tidak terelakan.



Gambar 1.1 Produk industri kreatif

Untuk disain motif, dibutuhkan suatu perangkat lunak yang menerjemahkan model atau motif yang diinginkan ke bahasa mesin bagi mesin CNC untuk mencetak obyek yang sesuai. Perangkat lunak tersebut umumnya berupa perangkat lunak *computer aided manufacturing* atau CAM yang pada umumnya berbayar. Oleh karena itu, dirancanglah perangkat lunak dalam tugas akhir ini untuk memenuhi kebutuhan tersebut.

1.2. Perumusan Masalah

Dalam tugas akhir yang berjudul “rancang bangun perangkat lunak untuk mengubah model 3 dimensi menjadi *G-code* bagi mesin CNC”

1. Cara agar mesin CNC dapat mencetak suatu benda sesuai dengan model 3 dimensi.
2. Perancangan perangkat lunak agar menghasilkan *g-code* bagi mesin CNC untuk mencetak benda yang akurat sesuai dengan model dalam durasi sesingkat mungkin.

1.3. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah :

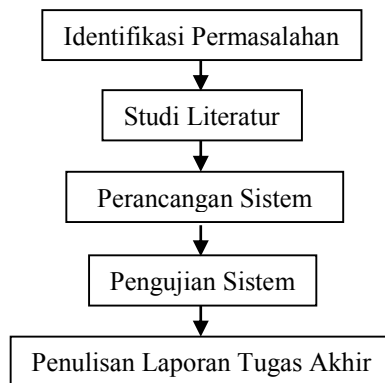
1. Sistem tidak dapat memerintahkan mesin CNC untuk berganti *tool* secara otomatis.
2. Sistem tidak dapat memerintahkan mesin CNC untuk mencetak obyek dengan model 3 dimensi yang memiliki *contour* (2.5 D).

1.4. Tujuan

Penelitian pada tugas akhir ini bertujuan untuk mencetak obyek sesuai dengan grafis komputer 2.5 dimensi sebagai *input* dari sistem.

1.5. Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini dapat dilihat pada gambar 1.1



Gambar 1.2 Metodologi pengerjaan tugas akhir

IDENTIFIKASI PERMASALAHAN

Pada tahapan ini penulis mengidentifikasi masalah yang bersangkutan dengan grafis komputer 3 dimensi, *G-code*, dan mesin CNC.

STUDI LITERATUR

Hal-hal yang dipelajari dalam tahap studi literatur, meliputi :

1. Mempelajari grafis komputer 3 dimensi.
2. Mempelajari kode pemrograman CNC yaitu *G-code*.
3. Mempelajari cara kerja mesin CNC.

PERANCANGAN SISTEM

Sistem yang dirancang merupakan sebuah perangkat lunak dengan *user interface* yang dirancang menggunakan sebuah *integrated development environment*. Perangkat lunak yang dirancang mengubah suatu *file* grafis komputer 3 dimensi menjadi sebuah *file* yang berisikan kode *G-code* bagi mesin CNC untuk mencetak obyek.

PENGUJIAN SISTEM

Ada beberapa pengujian yang dilakukan dalam tugas akhir ini, yaitu :

1. Pengujian *G-code* pada mesin CNC
2. Pengujian ukuran material cetak yang berbeda
3. Pengujian model yang berbeda

PENULISAN LAPORAN TUGAS AKHIR

Tahap ini adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Tahap ini dimulai saat pengambilan data. Laporan tugas akhir ini berisi tentang semua kegiatan yang dilakukan selama mengerjakan tugas akhir.

1.6. Sistematika Penulisan

Dalam penulisan buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Pada bagian ini menjelaskan tentang dasar penyusunan tugas akhir yang terdiri dari beberapa sub bagian, yaitu latar belakang, permasalahan dan

batasannya, tujuan yang diharapkan, metodologi pengerjaan tugas akhir, relevansi serta sistematika penulisan tugas akhir ini.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bagian ini menjelaskan tentang metode-metode pengenalan wajah yang pernah diteliti sebelumnya. Selain itu pada bagian ini juga dijelaskan dasar-dasar teori yang dibutuhkan dalam pengerjaan tugas akhir ini.

BAB III : PERANCANGAN SISTEM

Pada bagian ini berisi perancangan sistem yang membahas pemrograman perangkat lunak yang mengubah *file* grafis komputer 3 dimensi menjadi sebuah *G-code file*.

BAB IV : PENGUJIAN

Pada bagian ini akan menjelaskan mengenai cara kerja sistem dan analisisnya.

BAB V : PENUTUP

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan tugas akhir ini, serta saran-saran untuk pengembangannya.

1.7. Relevansi

Mata kuliah yang mendukung tugas akhir ini adalah robot industri, serta beberapa referensi tentang pemrograman menggunakan *integrated development environment*.

Hasil dari tugas akhir ini diharapkan dapat diimplementasikan pada Pusat Robotika Institut Teknologi Sepuluh Nopember dan laboratorium AJ 304.

BAB II

DASAR TEORI DAN TINJAUAN PUSTAKA

Dasar teori berisi tentang metode-metode yang telah ditemukan guna sebagai pembandingan dengan metode yang digunakan pada tugas akhir ini. Tinjauan pustaka dalam bab ini menjelaskan tentang sistem yang berkaitan dengan tugas akhir ini dan pernah diimplementasikan oleh peneliti-peneliti sebelumnya.

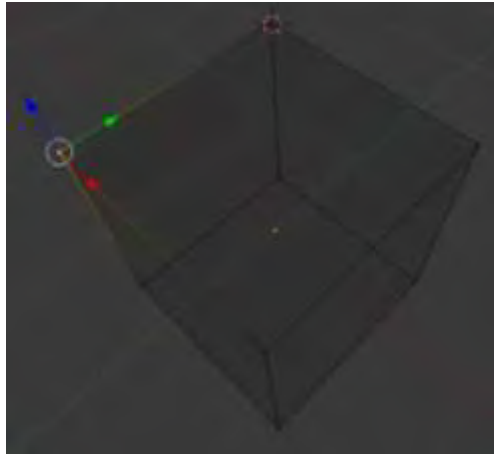
2.1. Model 3 Dimensi

Model 3 dimensi adalah grafis yang menggunakan representasi data geometris tiga dimensi yang disimpan dalam komputer untuk tujuan melakukan perhitungan dan *rendering* gambar 2 dimensi. Gambar tersebut dapat disimpan untuk dilihat kemudian atau ditampilkan secara *real-time*.

Pada dasarnya, model 3 dimensi terbentuk dari titik-titik yang disebut *vertex* yang menentukan dan membentuk *face*. *Face* adalah daerah yang terbentuk dari setidaknya tiga *vertex*. Sedangkan dua buah *vertex* akan membentuk garis yang disebut dengan *edge*.^[1]

2.1.1. Vertex

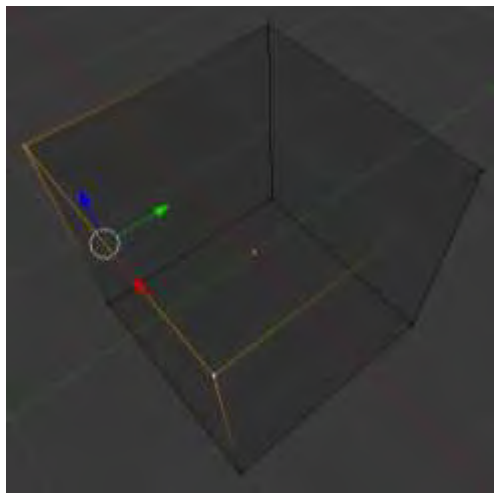
Sebuah *vertex* pada suatu grafis adalah objek, yang memiliki atribut tertentu, salah satunya adalah posisinya. Sebuah *vertex* adalah struktur data, yang menggambarkan atribut tertentu, di antaranya, posisi titik di ruang 2 dimensi atau 3 dimensi. Objek yang ditampilkan terdiri dari *array* permukaan datar dan *vertex* menentukan lokasi dan atribut lainnya dari sudut permukaan.^[1]



Gambar 2.1 *Vertex* pada sebuah kubus

2.1.2. *Edge*

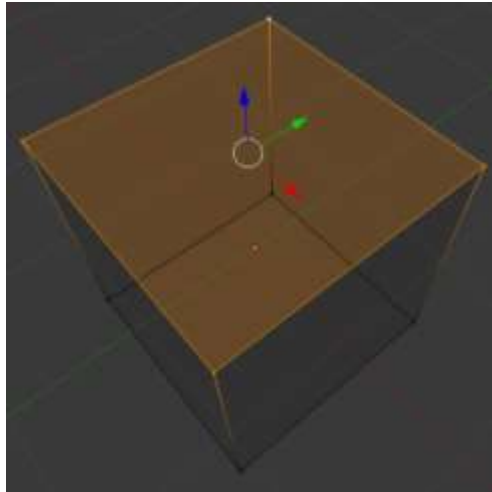
Sebuah *edge* pada suatu grafis adalah sebuah garis lurus yang dibentuk oleh dua buah *edge*. Sebuah *edge* juga dapat didefinisikan sebagai sebuah garis dimana dua buah *face* bertemu. ^[1]



Gambar 2.2 *Edge* pada sebuah kubus

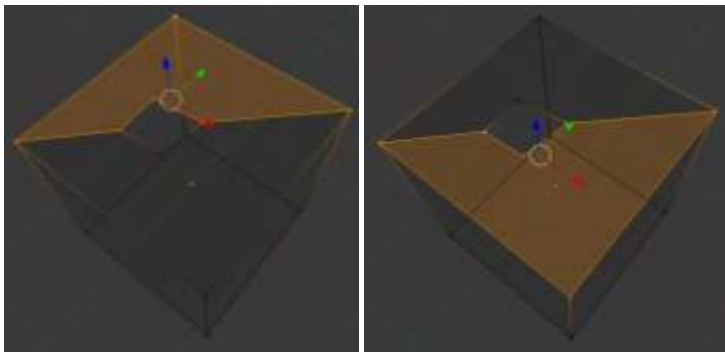
2.1.3. *Face*

Sebuah *face* pada suatu grafis adalah sebuah daerah yang dibentuk paling sedikit oleh tiga buah *vertex* atau tiga buah *edge*.^[1]



Gambar 2.3 *Face* pada sebuah kubus

Jika pada suatu *face* terdapat lubang, *face* tersebut akan berubah menjadi menjadi beberapa *face* tergantung pada jumlah lubang.



Gambar 2.4 *Face* dengan lubang.

2.1.4. Wavefront .obj File

Format *file* OBJ adalah format data sederhana yang mewakili geometri 3D sendiri - yaitu, posisi setiap titik, posisi UV dari setiap *vertex* koordinat tekstur, *vertex normal*, dan *face-face* yang membuat setiap poligon didefinisikan sebagai daftar *vertex*, dan *vertex* tekstur. *Vertex* disimpan dalam urutan berlawanan arah jarum jam secara *default*, sehingga deklarasi eksplisit dari *face normal* tidak dibutuhkan. Koordinat OBJ tidak memiliki unit, tetapi file OBJ dapat berisi informasi yang dapat dibaca pada baris komentar dengan karakter # di awal baris.^[2]

Berikut adalah contoh data pada *file* OBJ dari sebuah kubus:

```
# Blender v2.71 (sub 0) OBJ File: ''
# www.blender.org
mtllib cube.mtl
o Cube
v 1.000000 1.000000 -1.000000
v 1.000000 -1.000000 -1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 -1.000000
v 1.000000 0.999999 1.000000
v 0.999999 -1.000001 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 1.000000 1.000000
usemtl Material
s off
f 1 2 3 4
f 5 8 7 6
f 1 5 6 2
f 2 6 7 3
f 3 7 8 4
f 5 1 4 8
```

Baris pertama dan kedua dengan karakter '#' pada awal baris merupakan baris komentar yang menyediakan informasi bagi pembaca dan tidak memiliki pengaruh pada grafis 3dimensi.

```
# Blender v2.71 (sub 0) OBJ File: ''
# www.blender.org
```

Baris ketiga merupakan baris untuk menandakan *material library* yang digunakan. *Library* ini digunakan untuk kepentingan grafis dan pada tugas akhir ini, *library* tidak digunakan.

`mtllib cube.mtl`

Baris keempat menandakan objek dasar yang terdapat pada *file* dan ditandai dengan *header* berupa karakter ‘o’ pada awal baris.^[2]

o Cube

Baris kelima hingga kedelapan merupakan daftar *vertex* yang terdapat pada *file* dan ditandai dengan *header* berupa karakter ‘v’ pada awal baris. Data pada baris ini merupakan koordinat Cartesien X, Y, dan Z dari *vertex-vertex* yang terdapat pada *file*.^[2]

```
v 1.000000 1.000000 -1.000000
v 1.000000 -1.000000 -1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 -1.000000
v 1.000000 0.999999 1.000000
v 0.999999 -1.000001 1.000000
v -1.000000 -1.000000 1.000000
v -1.000000 1.000000 1.000000
```

Sebagai contoh, pada baris *vertex* pertama, *vertex* berada pada koordinat Cartesien (1, 1, -1).

```
v 1.000000 1.000000 -1.000000
```

Baris dengan *header* ‘usemtl’ menandakan nama material yang digunakan. Nama material yang digunakan tertera pada *material library* yang telah didefinisikan sebelumnya pada baris dengan *header* ‘mtllib’.^[2]

usemtl Material

Baris dengan *header* ‘s’ menandakan opsi *smooth shading* untuk menghaluskan objek. Baris ini menandakan opsi tersebut di nonaktifkan.^[2]

s off

Baris dengan *header* ‘f’ menandakan *face*. Data data setelah header merupakan indeks dari *vertex-vertex* yang telah didefinisikan

sebelumnya. Indeks *vertex* tidak tertera pada *file*, namun indeks merupakan urutan dari definisi *vertex*. Baris pertama dengan *header* 'v' merupakan *vertex* dengan indeks 1 dan karena *file* hanya berisi delapan *vertex*, baris terakhir dengan *header* 'v' merupakan *vertex* dengan indeks 8.^[2]

```
f 1 2 3 4
f 5 8 7 6
f 1 5 6 2
f 2 6 7 3
f 3 7 8 4
f 5 1 4 8
```

Sebagai contoh, baris pertama dengan *header* 'f' merupakan *face* yang dibentuk oleh empat *vertex* dengan indeks 1, 2, 3, dan 4 yang berarti *face* dibentuk oleh *vertex* dari baris pertama hingga baris keempat dengan *header* 'v'. *Vertex-vertex* tersebut adalah sebagai berikut :

```
v 1.000000 1.000000 -1.000000
v 1.000000 -1.000000 -1.000000
v -1.000000 -1.000000 -1.000000
v -1.000000 1.000000 -1.000000
```

2.2. G-Code

G-code adalah nama umum yang paling banyak digunakan sebagai bahasa pemrograman *numerical control* (NC). Hal ini digunakan terutama pada *computer-aided manufacturing* untuk mengendalikan peralatan mesin yang otomatis. *G-code* juga sering disebut sebagai bahasa pemrograman G.

Secara fundamental, *G-code* merupakan bahasa di mana orang memerintahkan alat mesin yang terkomputerisasi bagaimana membuat sesuatu. Bagaimana cara membuat sesuatu didefinisikan oleh petunjuk di mana untuk bergerak, seberapa cepat bergerak, dan melalui jalan apa untuk bergerak. Situasi paling umum adalah bahwa, dalam *tool* mesin, *tool* pemotong dipindahkan sesuai dengan instruksi ini melalui lintasan *tool* atau *toolpath* dan memotong sisa material untuk meninggalkan hanya benda kerja yang diinginkan.

Suatu *G-code* memiliki dua buah parameter utama, yaitu kode dan variabel.^[3]

2.2.1. Kode

Kode pada *G-code* dibagi menjadi dua, yaitu kode dasar dan kode lanjut.

Kode dasar pada *G-code* merupakan kode untuk pergerakan mesin dasar. Beberapa kode tersebut antara lain :

- G00
Kode ini berfungsi untuk menggerakkan *tool* secara lurus sesuai dengan variabel koordinat yang ditentukan dengan kecepatan gerak secepat mungkin.
- G01
Kode ini berfungsi sama dengan kode G00, hanya saja dengan kecepatan yang dapat diatur sesuai dengan variabel kecepatan.
- G02
Kode ini berfungsi untuk menggerakkan kode secara melingkar searah jarum jam.
- G03
Kode ini berfungsi sama dengan kode G02, hanya saja dengan arah yang berlawanan, yaitu berlawanan arah jarum jam.

Kode lanjut pada *G-code* adalah kode untuk fungsi-fungsi yang lebih spesifik. Beberapa kode tersebut antara lain :

- G70
Kode ini berfungsi untuk mengubah satuan unit dari variabel menjadi inci.
- G71
Kode ini berfungsi untuk mengubah satuan unit dari variabel menjadi milimeter.
- G90
Kode ini berfungsi untuk mengubah sistem koordinat menjadi sistem koordinat absolut.
- G91
Kode ini berfungsi untuk mengubah sistem koordinat menjadi sistem koordinat *incremental*.^[3]

2.2.2. Variabel

Variabel pada *G-code* merupakan parameter dengan nilai yang menentukan bagaimana kode berperilaku.

Beberapa variabel dasar pada *G-code* antara lain:

- X
Variabel ini mewakili posisi pada koordinat Cartesians X.
Variabel ini digunakan pada kode G00 dan G01.
- Y
Variabel ini mewakili posisi pada koordinat Cartesians X.
Variabel ini digunakan pada kode G00 dan G01.
- Z
Variabel ini mewakili posisi pada koordinat Cartesians X.
Variabel ini digunakan pada kode G00 dan G01.
- F
Variabel ini mewakili kecepatan gerak tool atau *feed rate* dari mesin. Nilai dari variabel ini dinyatakan dalam satuan mm per detik atau inch per detik sesuai dengan *program*. Variabel ini digunakan pada kode G01. ^[3]

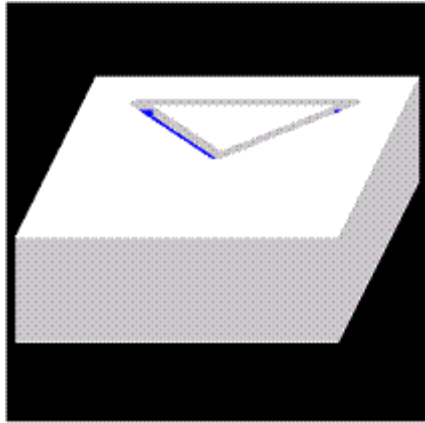
2.2.3. Format

Setiap baris pada *G-code file* diawali dengan kode, lalu variabel. Variabel yang telah ditentukan pada suatu baris akan berlaku pada baris dibawahnya jika tidak di definisikan ulang.

Pada baris pertama dan terakhir dari *file*, harus terdapat karakter ‘%’ sebagai tanda awal dan akhir *file*.

Berikut adalah contoh program untuk menggerakkan *tool* untuk membentuk sebuah segitiga.

```
%  
G01 X 1.500 Y 1.278 F 15  
G01 Z -0.200 F 5  
G01 X 1.693 Y 1.611 F 15  
G01 X 1.307 F 15  
G01 X 1.500 Y 1.278 F 15  
%
```



Gambar 2.5 Hasil dari *G-code* segitiga

Baris pertama dari *G-code* segitiga diatas menginstruksikan *tool* untuk bergerak ke koordinat Cartesian (1.5, 1.278, 0) dengan kecepatan atau *feed rate* maksimal. Baris kedua menginstruksikan *tool* untuk diturunkan secara perlahan sebanyak 0.2, atau dengan kata lain *tool* bergerak ke koordinat Cartesian (1.5, 1.278, -0.2) sehingga *tool* menembus permukaan obyek. Baris ketiga menginstruksikan *tool* untuk bergerak ke koordinat Cartesian (1.693, 1.611, -0.2) sehingga *tool* mengikis permukaan obyek. Lalu, baris keempat menginstruksikan *tool* untuk bergerak ke koordinat Cartesian (1.307, 1.611, -0.2) dan baris kelima menginstruksikan *tool* untuk bergerak kembali ke koordinat Cartesian awal (1.5, 1.278, -0.2) sehingga bentuk segitiga terkikis pada permukaan obyek dengan kedalaman 0.2. ^[4]

2.3. Perangkat Lunak ‘Blender’

Blender merupakan sebuah perangkat lunak animasi 3 dimensi yang tidak berbayar dan *open source*. Perangkat lunak mendukung segala proses yang terkait dengan animasi 3 dimensi, yaitu *modeling*, *rigging*, *animasi*, simulasi, *rendering*, *compositing*, dan *motion tracking*.

Blender adalah perangkat lunak *cross-platform* yang dapat beroperasi pada sistem operasi Linux, Windows, dan Macintosh. *Interface* menggunakan OpenGL untuk menyediakan kemudahan bagi pengguna.



Gambar 2.6 Logo Blender

Pada tugas akhir ini, perangkat lunak Blender digunakan untuk menghasilkan suatu grafis komputer 3 dimensi dalam bentuk *file* dengan ekstensi *.obj* yang digunakan untuk pengujian data. Perangkat lunak Blender digunakan dalam tugas akhir ini karena tidak berbayar dan proses pembangunan suatu model 3 dimensi yang relatif mudah.^[5]

2.4. Perangkat Lunak ‘OpenSCAM’

OpenSCAM adalah perangkat lunak yang bersifat *open-source* dan tidak berbayar yang digunakan untuk simulasi proses *milling* atau *engraving* dari mesin CNC dengan 3 *axis*.



Gambar 2.7 Logo OpenSCAM

Dalam tugas akhir ini, perangkat lunak OpenSCAM digunakan untuk simulasi *file .gcode* yang dihasilkan, sebelum diberikan pada mesin CNC, oleh perangkat lunak yang dirancang untuk keperluan pengujian data. Perangkat lunak OpenSCAM digunakan karena tidak berbayar dan proses simulasi yang sederhana.^[6]

2.5. Perangkat Lunak ‘Lazarus’

Lazarus merupakan *visual integrated development environment* (IDE) tidak berbayar yang digunakan untuk *rapid application development* (RAD) menggunakan bahasa pemrograman dan *compiler* Free Pascal.



Gambar 2.8 Lazarus

Pada tugas akhir ini, Lazarus digunakan untuk merancang perangkat lunak yang digunakan untuk mengubah grafis komputer 3 dimensi, dalam bentuk *file* .obj yang dihasilkan oleh perangkat lunak Blender, menjadi *file* dengan ekstensi .gcode, yang akan disimulasikan pada perangkat lunak OpenSCAM, yang akan diberikan pada mesin CNC untuk mencetak obyek sesuai dengan grafis komputer 3 dimensi.^[7]

2.6. Perangkat Lunak ‘Mastercam’

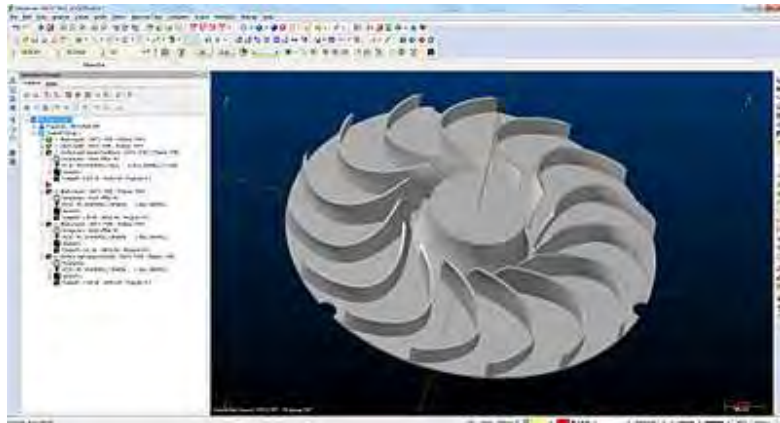
Mastercam adalah sistem *computer aided manufacturing* atau CAM yang dibuat dan dikembangkan oleh CNC Software, Inc. Mastercam dilengkapi dengan fitur *computer aided design* atau CAD yang memungkinkan pengguna melakukan disain pada layar komputer seperti pada gambar 2.10.



Gambar 2.9 Logo Mastercam

Mastercam memiliki seperangkat jalur pergerakan perkakas atau *toolpath* yang telah terdefiniskan yang memungkinkan proses, seperti

pocketing, engraving, drilling, face peel milling, dan masih banyak lagi sehingga obyek cetak dapat dihasilkan secara efisien dan akurat.



Gambar 2.10 *User Interface* dari Mastercam

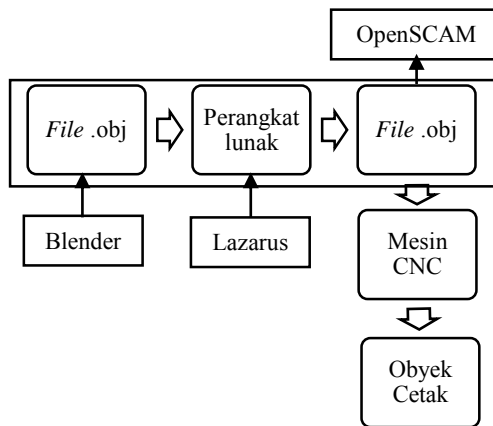
Dengan versi yang terbaru, Mastercam X beralih sepenuhnya pada sistem operasi Windows dari sistem operasi DOS. Selain itu, Mastercam X memiliki fitur-fitur baru seperti *solid modeling*, *4-axis machining*, dan *5-axis machining*.

Jika dibandingkan dengan perangkat lunak yang dirancang pada tugas akhir ini, Mastercam dapat mencetak obyek sesuai dengan model 3 dimensi yang memiliki kontur yang dapat dihasilkan melalui CAD pada MasterCAM atau dari CAD lain. Selain itu, jalur pergerakan perkakas atau *toolpath* yang dihasilkan sangat efisien dan tidak bergantung pada bagaimana karakteristik dan parameter model didefinisikan.^[8]

BAB III

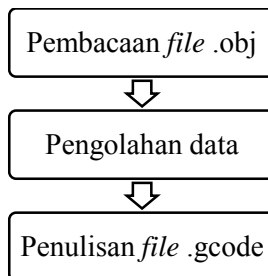
PERANCANGAN SISTEM

Dalam pembuatan tugas akhir ini, sebuah perangkat lunak dirancang untuk membaca grafis komputer 3 dimensi dalam bentuk *file* dengan *format* OBJ dan menulis sebuah *file* dengan *format* GCODE untuk diunduh pada mesin CNC. Secara garis besar keseluruhan sistem dapat dilihat pada gambar 3.1.



Gambar 3.1 Keseluruhan sistem

Perangkat lunak bekerja dalam tiga tahap utama. Gambar 3.2 menggambarkan bagaimana perangkat lunak bekerja.



Gambar 3.2 Proses perangkat lunak

3.1. Pembacaan *File .obj*

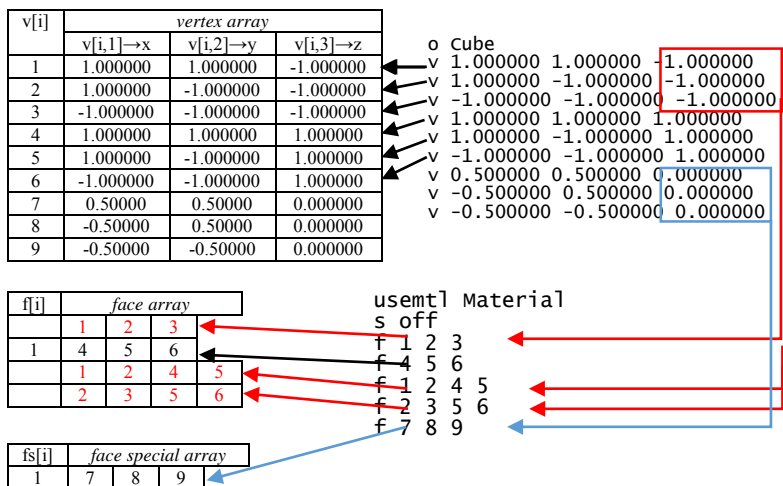
Karakteristik suatu model grafis komputer 3 dimensi dibaca melalui *file .obj*. *File* diunduh melalui *user interface* pada perangkat lunak yang dirancang. *File* yang telah diunduh ke dalam perangkat lunak akan dibaca setiap baris dan dibaca sebanyak dua kali, pertama untuk *vertex* dan kedua untuk *face*.

Pembacaan pertama merupakan pembacaan data *vertex*. Setiap baris dengan karakter awal 'v' menandakan bahwa baris tersebut mendefinisikan sebuah *vertex*, sehingga data berupa koordinat Cartesian x, y, dan z yang dipisahkan dengan karakter spasi pada *file .obj* disimpan dalam sebuah *vertex array*.

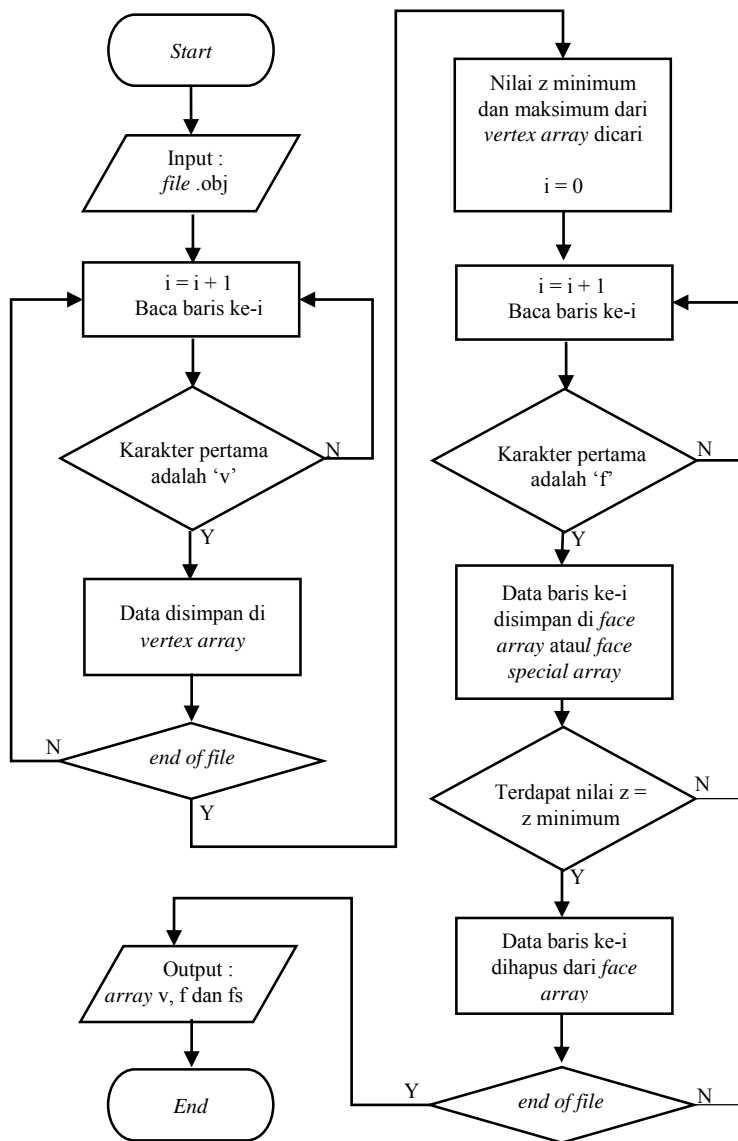
Setelah pembacaan pertama selesai, dicari nilai z minimum dan maksimum dari data *vertex* yang telah didapat. Hal ini dilakukan untuk memisahkan sisi atas dan sisi bawah dari grafis komputer.

Pembacaan kedua merupakan pembacaan data *face*. Setiap baris dengan karakter awal 'f' menandakan bahwa baris tersebut mendefinisikan sebuah *face*, sehingga data berupa indeks atau urutan definisi *vertex* yang dipisahkan dengan karakter spasi pada *file .obj* disimpan dalam sebuah *face array*. Setiap kali suatu baris *face* dibaca, data *face* dikorelasikan dengan nilai z dari *vertex* yang indeksinya merupakan data dari *face* tersebut. Jika terdapat sebuah nilai z yang sama dengan nilai z minimum dari data *vertex* yang telah didapat, data yang baru saja dimasukkan ke *face array* dihapus. Jika terdapat sebuah nilai z yang terdapat diantara nilai z minimum dan nilai z maksimum, suatu *face* akan dimasukkan ke *face special array*.

Ilustrasi dari proses ini dapat dilihat pada gambar 3.3 dan diagram alir dapat dilihat pada gambar 3.4.



Gambar 3.3 Ilustrasi pembacaan file .obj



Gambar 3.4 Diagram alir proses pembacaan *file .obj*

Pembacaan kedua merupakan pembacaan data *face*. Setiap baris dengan karakter awal ‘f’ menandakan bahwa baris tersebut mendefinisikan sebuah *face*, sehingga data berupa indeks atau urutan definisi *vertex* yang dipisahkan dengan karakter spasi pada *file* .obj disimpan dalam sebuah *face array*. Setiap kali suatu baris *face* dibaca, data *face* dikorelasikan dengan nilai z dari *vertex* yang indeksnya merupakan data dari *face* tersebut. Jika terdapat sebuah nilai z yang sama dengan nilai z minimum dari data *vertex* yang telah didapat, data yang baru saja dimasukkan ke *face array* dihapus.

3.2. Pengolahan Data

Pada tahap pengolahan data, terdapat tiga bagian proses pengolahan data, yaitu pengolahan *vertex*, pengolahan *face*, dan pengolahan akhir.

3.2.1. Pengolahan *Vertex*

Dalam pengolahan *vertex*, dilakukan pergeseran nilai dan penskalaan. Pertama, setiap anggota *vertex array* diubah menjadi nilai positif. Hal ini dilakukan dengan menggeser nilai setiap anggota *vertex array* dengan mencari nilai x, dan y terkecil dan mengurangi setiap anggota *vertex array* dengan nilai-nilai minimum yang didapat. Selain itu, nilai z untuk sementara waktu diganti dengan nilai nol. Ilustrasi dari proses ini dapat dilihat pada gambar 3.5.

v[i]	vertex array		
	v[i,1]→x	v[i,2]→y	v[i,3]→z
1	-1.0	-2.0	-2.0
2	-1.0	1.0	-2.0
3	1.0	-1.0	-2.0
4	-1.0	-2.0	2.0
5	-1.0	1.0	2.0
6	1.0	-1.0	2.0

↓

v[i]	vertex array		
	v[i,1]→x	v[i,2]→y	v[i,3]→z
1	0.0	0.0	0.0
2	0.0	3.0	0.0
3	2.0	1.0	0.0
4	0.0	0.0	0.0
5	0.0	3.0	0.0
6	2.0	1.0	0.0

Gambar 3.5 Ilustrasi pergeseran nilai menjadi positif

Lalu, setiap anggota *vertex array* diperbesar sesuai dengan ukuran obyek cetak, panjang dan lebar, yang didefinisikan melalui *user interface*. Nilai x dan y maksimum dicari, lalu masing-masing ukuran obyek dibagi dengan masing-masing nilai maksimum, nilai panjang dibagi dengan nilai x maksimum dan nilai lebar dibagi dengan nilai y maksimum. Masing-masing hasil pembagian kemudian saling dibandingkan. Hasil pembagian yang paling kecil dijadikan skala dan dikalikan pada setiap anggota dari *vertex array*. Ilustrasi dari penskalaan model dapat dilihat pada gambar 3.6.

v[i]	vertex array		
	v[i,1]→x	v[i,2]→y	v[i,3]→z
1	0.0	0.0	0.0
2	0.0	3.0	0.0
3	2.0	1.0	0.0
4	0.0	0.0	0.0
5	0.0	3.0	0.0
6	2.0	1.0	0.0

Panjang (x) = 10

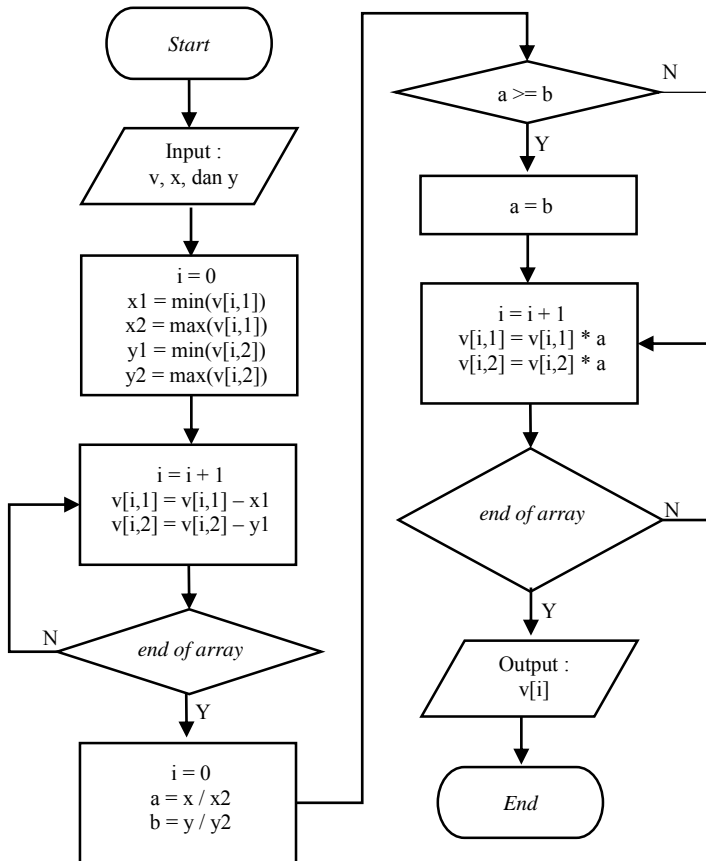
Lebar (y) = 10

$$\frac{10}{2} > \frac{10}{3}$$

v(i)	vertex array		
	v[i,1]→x	v[i,2]→y	v[i,3]→z
1	0.0	0.0	0.0
2	0.0	10.0	0.0
3	6.666667	3.333334	0.0
4	0.0	0.0	0.0
5	0.0	10.0	0.0
6	6.666667	3.333334	0.0

Gambar 3.6 Penskalaan model

Keseluruhan proses pengolahan *vertex*, dapat dilihat pada diagram alir pada gambar 3.7. Input dari diagram alir merupakan *vertex array* v[i] dan ukuran obyek cetak, yaitu panjang x dan lebar y. Pada diagram alir terdapat proses pencarian nilai x minimum x1 dan maksimum x2, serta nilai y minimum y1 dan maksimum y2. Output dari diagram alir merupakan *vertex array* v.



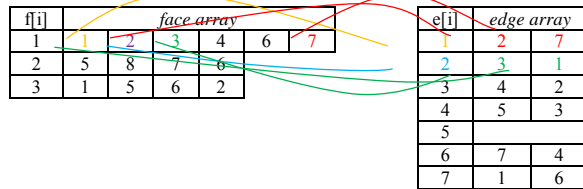
Gambar 3.7 Pengolahan *vertex*

3.2.2. Pengolahan *Face*

Dalam pengolahan *face*, karakteristik dari model 3 dimensi yang tidak terdefiniskan pada *file .obj* ditentukan. Karakteristik tersebut adalah *edge*.

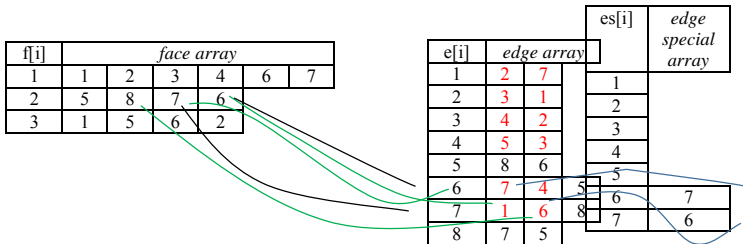
Edge diperoleh dengan membaca data tiap *face* pada *face array*. Tiap dua buah data, yang berupa indeks untuk *vertex array*, yang

bersebelahan pada tiap *face* diletakkan pada sebuah *edge array*. Ilustrasi untuk proses ini dapat dilihat pada gambar 3.8.



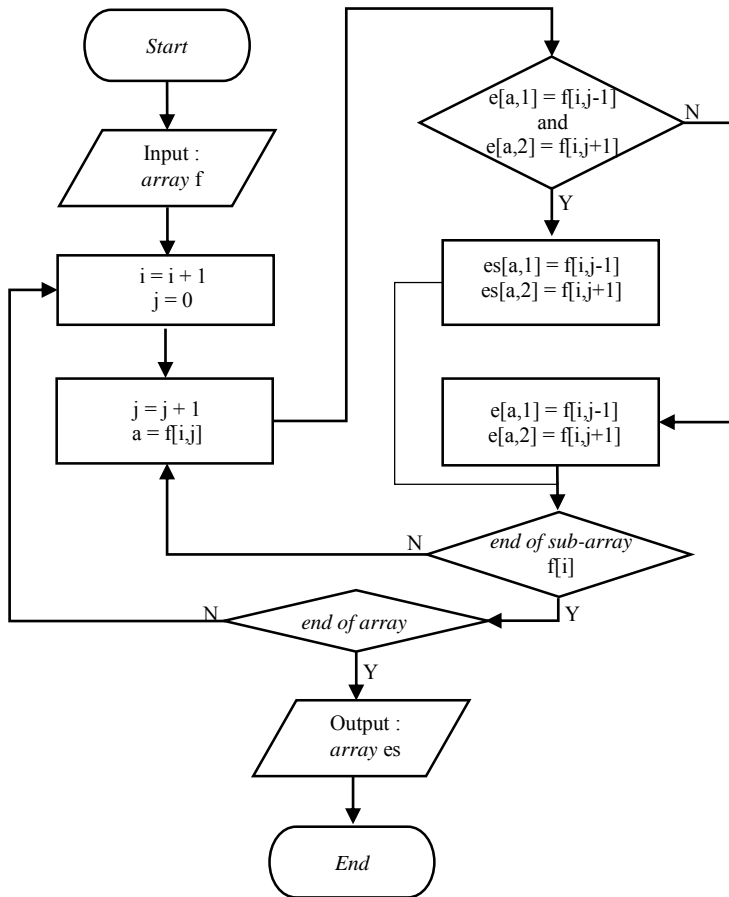
Gambar 3.8 Ilustrasi dalam mendapatkan *edge*

Jika terdapat dua *edge* yang sama pada dua buah *face*, *edge* tersebut dikategorikan sebagai *special edge* dan diletakkan bukan di *edge array*, melainkan diletakkan pada *edge array* khusus, *edge special array*. *Edge special array* akan digunakan pada proses pengolahan selanjutnya, yaitu pengolahan akhir. Ilustrasi pada proses ini dapat dilihat pada gambar 3.9.



Gambar 3.9 Ilustrasi dua *edge* yang sama pada dua *face* yang berbeda

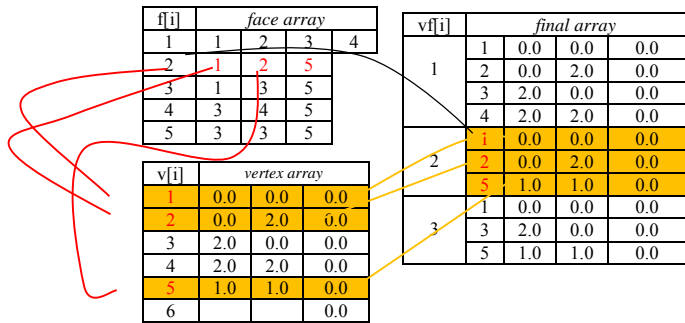
Diagram alir dari proses pengolahan *face* dapat dilihat pada gambar 3.10. Input dari diagram alir merupakan *face array* *f* dan output dari diagram alir merupakan *edge special array* *es*.



Gambar 3.10 Diagram alir proses pengolahan *face*

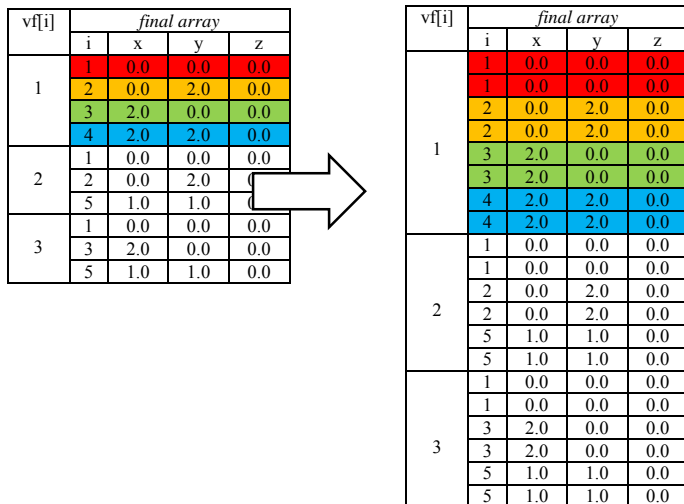
3.2.3. Pengolahan Akhir

Dalam pengolahan akhir, data dari *vertex array* dan *face array* dipindahkan ke suatu *array* akhir, *final array* yang merupakan *array* dengan isi berupa *sub-array*. Ilustrasi dari proses ini dapat dilihat pada gambar 3.11.



Gambar 3.11 Ilustrasi pemindahan *vertex* ke *array* akhir

Dari *array* akhir yang telah diperoleh, dilakukan pengolahan data kembali. Data-data berupa *vertex* pada *final array* digandakan seperti pada ilustrasi pada gambar 3.12.



Gambar 3.12 Ilustrasi penggandaan *array*

Selanjutnya, dilakukan perubahan nilai *z* pada setiap *vertex* pada *final array*. Jika dalam *final array* terdapat *edge* yang terletak pada *array edge* khusus, nilai *z* dari kedua *vertex* pada *edge* tersebut yang

semula bernilai nol diubah menjadi suatu nilai yang bernilai positif yang menandakan agar *tool* dari mesin CNC diangkat agar tidak mengikis obyek.

es[i]	Edge special array	vf[i]	Final array			
			$v[i,j,1] \rightarrow i$	$v[i,j,1] \rightarrow x$	$v[i,j,2] \rightarrow y$	$v[i,j,3] \rightarrow z$
1			1			-5.0
2	4		1			-5.0
3	6		2			-5.0
4	2		2			5.0
5			4			5.0
6	3		4			-5.0
7			7			-5.0
8			7			-5.0
9			8			-5.0
10			8			-5.0
		7	3			-5.0
			3			5.0
			6			5.0
			6			-5.0
			5			-5.0
			5			-5.0
			9			-5.0
			9			-5.0
			10			-5.0
			10			-5.0

Gambar 3.13 Ilustrasi perubahan nilai z

Setelah itu, *final array* disederhanakan dengan menghapus salah satu dari dua *vertex* atau anggota yang sama. Ilustrasi dari proses penyederhanaan dapat dilihat pada gambar 3.14.

vf[i]	final array			
	$v[i,j,1] \rightarrow i$	$v[i,j,1] \rightarrow x$	$v[i,j,2] \rightarrow y$	$v[i,j,3] \rightarrow z$
7	1			-5.0
	1			-5.0
	2			-5.0
	2			5.0
	4			5.0
	4			-5.0
	7			-5.0
	7			-5.0
	8			-5.0
	8			-5.0
	3			-5.0
	3			5.0
	6			5.0
	6			-5.0
	5			-5.0
	5			-5.0
	9			-5.0
	9			-5.0
	10			-5.0
	10			-5.0

Gambar 3.14 Ilustrasi proses penyederhanaan *final vertex*

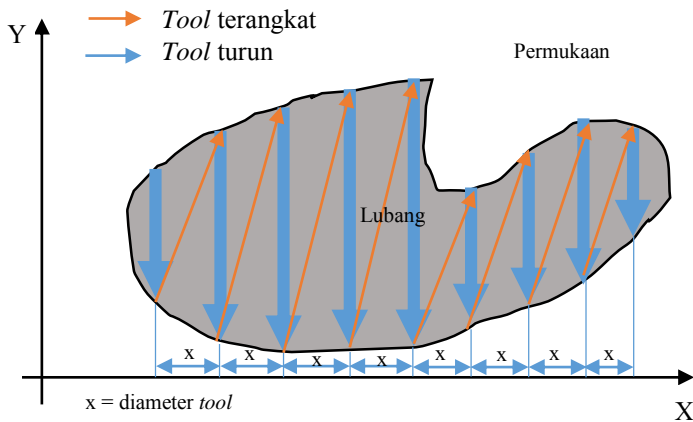
Setelah *final array* disederhanakan, akan ditambahkan data berupa *face* dari *special face array*. Data yang ditambahkan memiliki nilai z diantara nilai z minimum pada *final array* dan nilai nol. Hal ini dilakukan untuk melakukan proses *embossing*. Jika tidak terdapat *face* pada *special face array*, maka tidak ada data yang ditambahkan dan proses akan menjadi proses *cutting*.

vf[i]	final array			
	$v[i,j,1] \rightarrow i$	$v[i,j,1] \rightarrow x$	$v[i,j,2] \rightarrow y$	$v[i,j,3] \rightarrow z$
	1			-5.0
	2			-5.0
	2			5.0
	4			5.0
	4			-5.0
	7			-5.0
	8			-5.0
	3			-5.0
	3			5.0
	6			5.0
	6			-5.0
	5			-5.0
	9			-5.0
	10			-5.0
	11			-2.5
	12			-2.5
	13			-2.5

fs[i]	face special array		
1	11	12	13

Gambar 3.15 Ilustrasi proses penambahan data *final vertex*

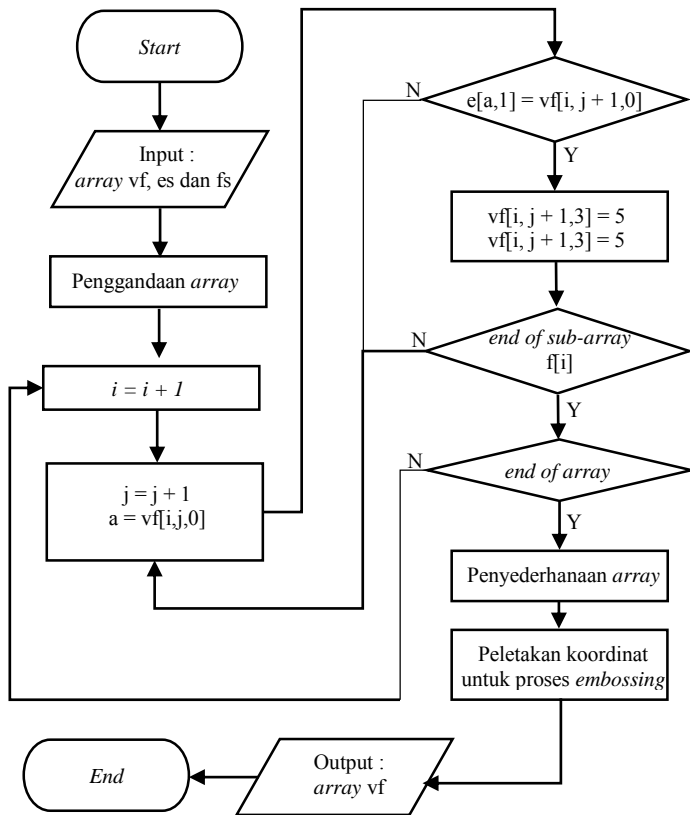
Proses penghilangan lubang pada proses *embossing* dilakukan melalui pergerakan *tool* yang seragam pada tiap *face* yang terdapat pada *face special array*. Pergerakan *tool* yang seragam tersebut merupakan pergerakan lurus pada tiap koordinat x yang tetap pada tiap *face* yang terdefinisi pada *face special array*. Agar lebih jelas, proses ini dapat dilihat pada gambar 3.16.



Gambar 3.16 Proses *embossing* pada suatu lubang pada permukaan

Koordinat-koordinat yang terdapat pada proses *embossing* pada gambar 3.16 kemudian diletakkan pada *final array*.

Diagram alir dari pengolahan akhir dapat dilihat pada gambar 3.17. Input dari diagram alir merupakan *edge special array* dan *final array*. Sedangkan output dari diagram alir merupakan *final array*.



Gambar 3.17 Diagram alir proses pengolahan akhir

3.3. Penulisan File bagi Mesin CNC

File yang diberikan ke mesin CNC, dengan ekstensi .gcode, ditulis perbaris berdasarkan *final array* yang telah dibuat dan diolah. Sebelum ditulis berdasarkan data pada *final array*, *file* ditulis kode-kode dasar seperti sistem koordinat absolut G90, satuan gerak milimeter G21, kecepatan putar *tool* M3, dan tanda awal program %.

File .gcode ditulis berdasarkan *sub-array* pada *final array*. Jika suatu *sub-array* habis, *tool* diangkat sebelum berlanjut ke *sub-array* berikutnya.

Proses dari penulisan *file* .gcode dapat dilihat pada ilustrasi pada gambar 3.16.

vf(i)	final array			
1	1	0.0	0.0	-5.0
	2	0.0	2.0	5.0
	3	2.0	0.0	5.0
	4	2.0	2.0	-5.0
2	1	0.0	0.0	-5.0
	2	0.0	2.0	5.0
	5	1.0	1.0	5.0

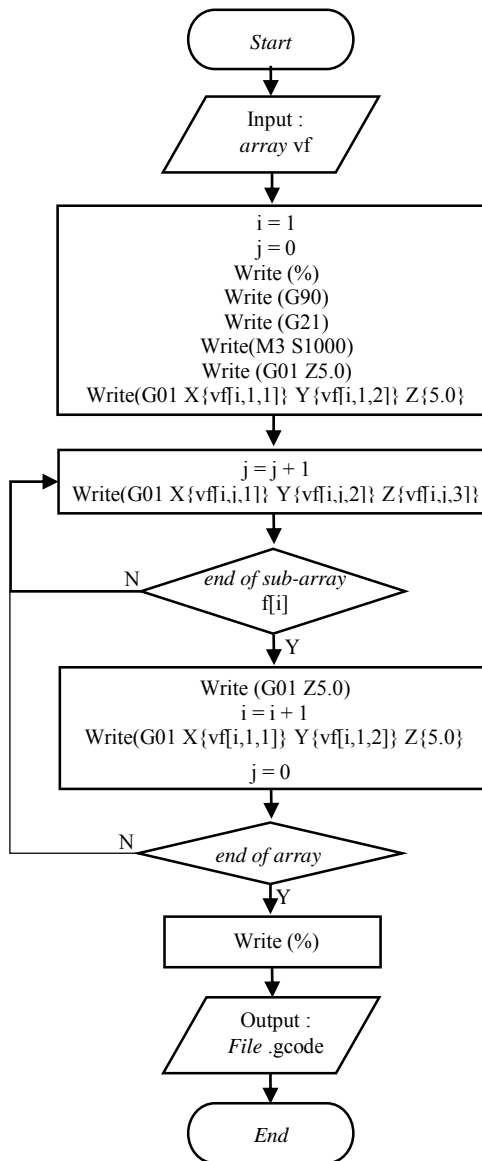
```

%
G90
G21
M3 S1000
G00 Z5.0
G01 X0.0 Y0.0 Z5.0
G01 X0.0 Y0.0 Z-5.0
G01 X0.0 Y2.0 Z5.0
G01 X2.0 Y0.0 Z5.0
G01 X2.0 Y2.0 Z-5.0
G00 Z5.0
G01 X0.0 Y0.0 Z5.0
G01 X0.0 Y0.0 Z-5.0
G01 X0.0 Y2.0 Z5.0
G01 X1.0 Y1.0 Z5.0

```

Gambar 3.18 Penulisan *file* .gcode

Diagram alir dapat dilihat pada gambar 3.17. Input dari diagram alir merupakan *final array* vf. Sedangkan output dari diagram alir merupakan *file* .obj. Setiap baris atau anggota pada *final array*, akan dilakukan penulisan baris pada *file* .gcode. Data pada *final array* yang memiliki tipe data numerik akan ditulis sebagai data dengan tipe data *string* atau sebagai karakter. Hal ini dapat dilakukan dengan *pseudo-code* Write dan kode Writeln pada Lazarus. Setiap baris akan ditulis kode-kode pada bahasa G-code yang kemudian akan diikuti oleh variabel-variabel yang nilai-nilainya merupakan nilai-nilai pada *final array*.



Gambar 3.19 Diagram alir penulisan *file .gcode*

BAB IV PENGUJIAN DATA

Terdapat beberapa data yang digunakan untuk menguji sistem. Data yang digunakan berupa grafis komputer 3 dimensi yang berupa pola sederhana dan rumit serta gambar atau portret. Pengujian data dibagi menjadi dua proses, yaitu proses *cutting* atau pemotongan dan proses *embossing*.

Data-data yang diuji menggunakan proses *cutting* dikategorikan dalam tiga kategori berdasarkan kompleksitas dan waktu proses mesin CNC, yaitu model sederhana, model dengan tingkat kerumitan sedang, dan model kompleks. Sedangkan data-data yang diuji menggunakan proses *embossing* tidak memiliki kategori.

4.1. Perangkat Keras yang Digunakan

Mesin CNC yang digunakan merupakan mesin CNC *router* Excitech SHG 1224 dengan spesifikasi pada tabel 4.1.

<i>Spindle</i>	1.5 KW Chinese Spindle
Kontroler	NC ARM Hand Held
<i>Switch</i>	Japan OMRON Switch
Luas area kerja	1220*2440*120mm
Ukuran meja kerja	1200*2400mm
Transmisi	X/Y/Z ball screw drive
Struktur meja kerja	T-slot
Daya <i>spindle</i>	1.5KW self cooling spindle
Kecepatan <i>spindle</i>	24000 rpm
Kecepatan	Kecepatan jelajah maksimum: 6000 mm/min
	Kecepatan kerja maksimum: 5000 mm/min
Driving system	Stepper
Tegangan kerja	AC220V/1PH/50HZ/60HZ
Sistem operasi	NC ARM Hand Held
Berat bersih	600KG
Berat kotor	800KG

Tabel 4.1 Spesifikasi Excitech SHG 1224^[9]

Proses pencetakan obyek menggunakan *tool* dengan diameter 2 milimeter. Kecepatan pergerakan *tool* atau *feedrate* diatur pada 1200 milimeter per menit, atau 2 centimeter per detik dan kecepatan putar *tool* atau *spindle speed* diatur menjadi 10.000 rpm. Parameter-parameter tersebut diatur dengan nilai-nilai tersebut untuk menekan resiko rusaknya *tool* menjadi sekecil mungkin.

Untuk proses *cutting*, digunakan material berbahan *medium-density fibreboard* atau MDF dengan ukuran 30 cm x 30 cm. Sedangkan untuk proses *embossing*, digunakan material berbahan *acrylic* dengan ukuran 10 cm x 10 cm. Bahan dan ukuran dari material yang digunakan pada kedua proses tidak sama dengan tujuan untuk mempersingkat waktu dan menghasilkan obyek yang lebih jelas pada proses *embossing*.

4.2. Model Sederhana

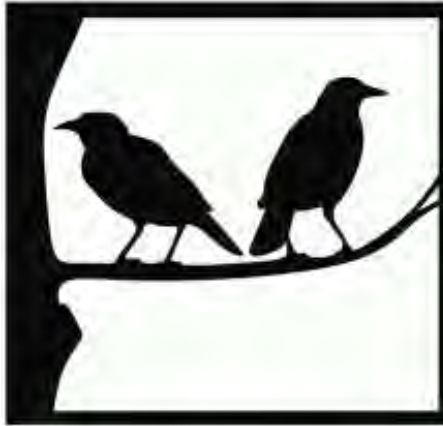
Model-model berikut dikategorikan dalam model sederhana karena waktu yang dibutuhkan oleh mesin CNC untuk mencetak model-model berikut relatif singkat.

4.2.1. Data Pertama

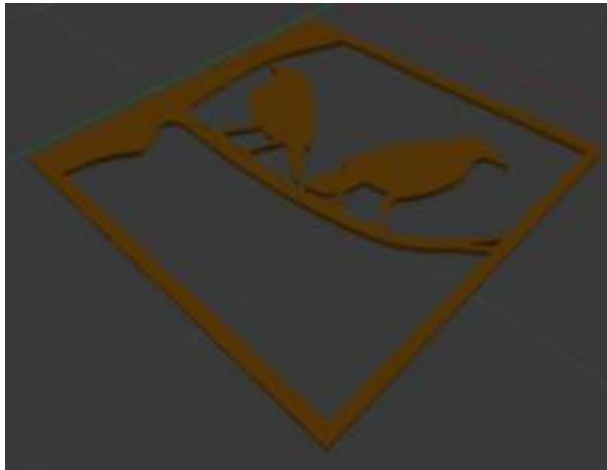
Data pertama dari kategori model sederhana merupakan bayangan dari sepasang burung seperti pada gambar 4.1. Grafis komputer 3 dimensi seperti pada gambar 4.2 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file .gcode* yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.3.

Face dan *vertex* dari grafis komputer pada gambar 4.2 yang terdefiniskan pada *file .obj* berjumlah 6 buah *face* dan 204 buah *vertex*.

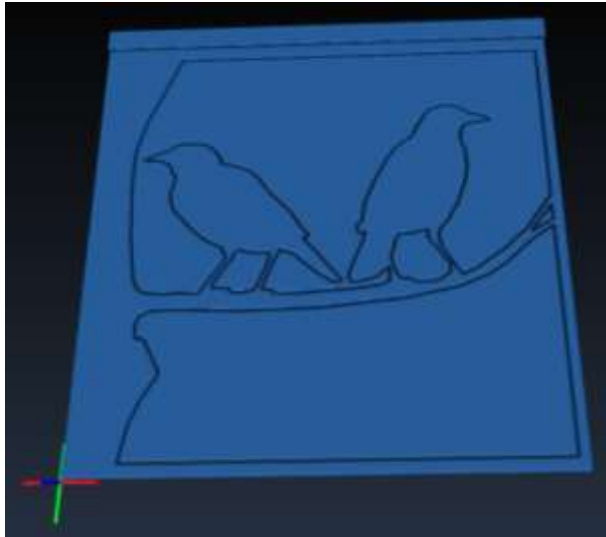
Ketika *file .gcode* yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.4. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 5 menit dan 38 detik.



Gambar 4.2 Gambar 2 dimensi dari data pertama



Gambar 4.3 Model 3 dimensi dari data pertama



Gambar 4.4 Hasil simulasi dari data pertama



Gambar 4.5 Obyek yang dihasilkan mesin CNC

4.2.2. Data Kedua

Data kedua dari kategori model sederhana merupakan logo dari Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember seperti pada gambar 4.5. Grafis komputer 3 dimensi seperti pada gambar 4.6 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file .gcode* yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.7.

Face dan *vertex* dari grafis komputer pada gambar 4.6 yang terdefinisikan pada *file .obj* berjumlah 13 buah *face* dan 1755 buah *vertex*.

Ketika *file .gcode* yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.8. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 8 menit dan 46 detik.



Gambar 4.6 Gambar 2 dimensi dari data kedua



Gambar 4.7 Model 3 dimensi dari data kedua



Gambar 4.8 Hasil simulasi dari data kedua



Gambar 4.9 Obyek yang dihasilkan mesin CNC

4.2.3. Data Ketiga

Data ketiga dari kategori model sederhana merupakan logo dari Institut Teknologi Sepuluh Nopember seperti pada gambar 4.9. Grafis komputer 3 dimensi seperti pada gambar 4.10 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.11.

Face dan *vertex* dari grafis komputer pada gambar 4.10 yang terdefiniskan pada *file* .obj berjumlah 5 buah *face* dan 3320 buah *vertex*.

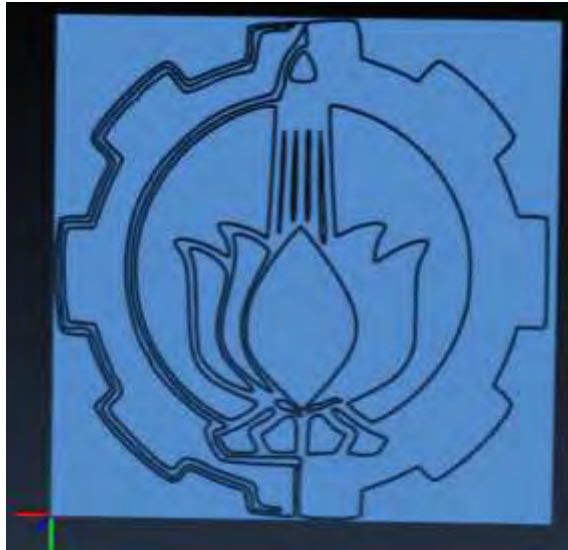
Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.12. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 12 menit dan 20 detik.



Gambar 4.10 Gambar 2 dimensi dari data ketiga



Gambar 4.11 Gambar 4.12 Model 3 dimensi dari data ketiga



Gambar 4.13 Hasil simulasi dari data ketiga



Gambar 4.14 Obyek yang dihasilkan mesin CNC

Karena ukuran yang terlalu kecil dan material yang rapuh, terdapat ketidaksesuaian antara obyek yang dihasilkan dengan grafik komputer 3 dimensi.

4.3. Model dengan Tingkat Kerumitan Sedang

Model- model berikut dikategorikan dalam model sederhana karena waktu yang dibutuhkan oleh mesin CNC untuk mencetak model-model berikut relatif cukup lama.

4.3.1. Data Pertama

Data pertama dari kategori model dengan tingkat kerumitan sedang merupakan bayangan atau siluet dari seseorang seperti pada gambar 4.13. Grafis komputer 3 dimensi seperti pada gambar 4.14 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.15.

Face dan *vertex* dari grafis komputer pada gambar 4.14 yang terdefiniskan pada *file* .obj berjumlah 3 buah *face* dan 8468 buah *vertex*.

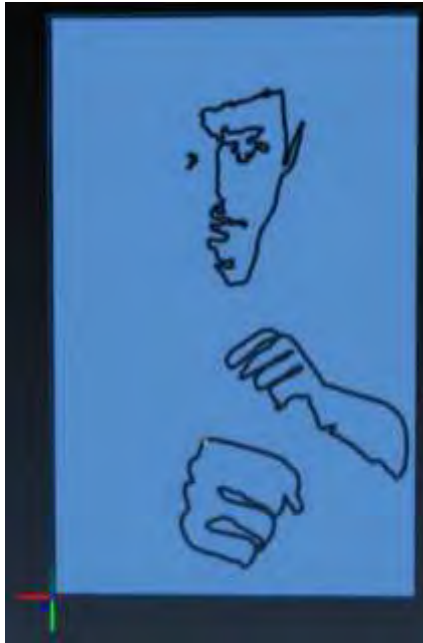
Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.16. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 20 menit dan 2 detik.



Gambar 4.15 Gambar 2 dimensi dari data pertama



Gambar 4.16 Model 3 dimensi dari data pertama



Gambar 4.17 Hasil simulasi dari data pertama



Gambar 4.18 Obyek yang dihasilkan mesin CNC

Karena ukuran yang terlalu kecil dan material yang rapuh, terdapat ketidaksesuaian antara obyek yang dihasilkan dengan grafik komputer 3 dimensi.

4.3.2. Data Kedua

Data kedua dari kategori model dengan tingkat kerumitan sedang merupakan suatu pola seperti pada gambar 4.17. Grafis komputer 3 dimensi seperti pada gambar 4.18 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.19.

Face dan *vertex* dari grafis komputer pada gambar 4.18 yang terdefiniskan pada *file* .obj berjumlah 19 buah *face* dan 12146 buah *vertex*.

Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.20. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 22 menit dan 28 detik.



Gambar 4.19 Gambar 2 dimensi dari data kedua



Gambar 4.20 Gambar 4.21 Model 3 dimensi dari data kedua



Gambar 4.22 Hasil simulasi dari data kedua



Gambar 4.23 Obyek yang dihasilkan mesin CNC

4.3.3. Data Ketiga

Data ketiga dari kategori model dengan tingkat kerumitan sedang merupakan suatu pola seperti pada gambar 4.21. Grafis komputer 3 dimensi seperti pada gambar 4.22 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.23.

Face dan *vertex* dari grafis komputer pada gambar 4.22 yang terdefinisikan pada *file* .obj berjumlah 32 buah *face* dan 10851 buah *vertex*.

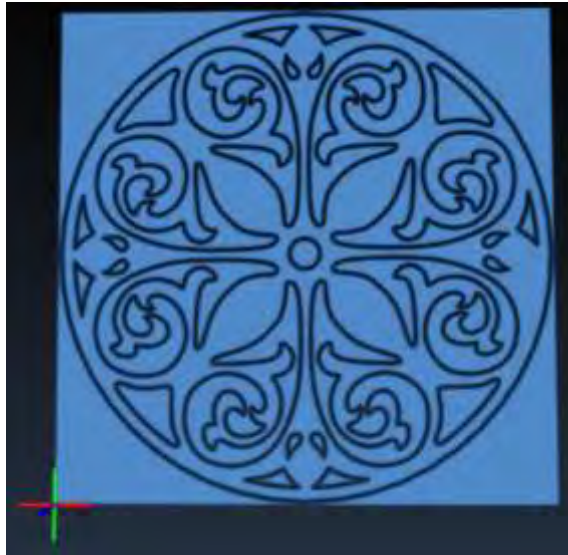
Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.24. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 25 menit dan 32 detik.



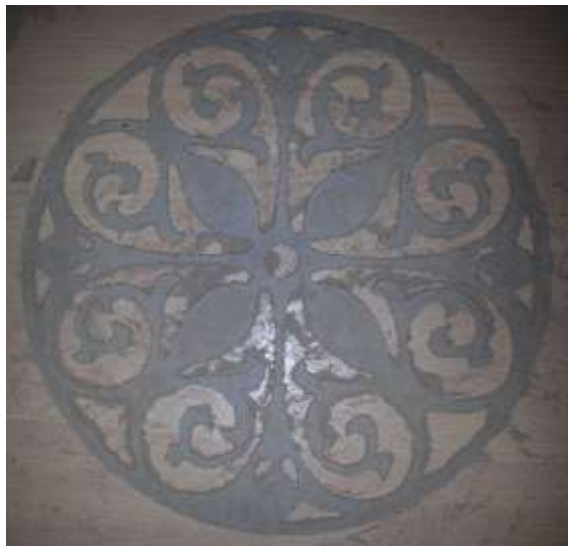
Gambar 4.24 Gambar 2 dimensi dari data ketiga



Gambar 4.25 Gambar 4.26 Model 3 dimensi dari data ketiga



Gambar 4.27 Hasil simulasi dari data ketiga



Gambar 4.28 Obyek yang dihasilkan mesin CNC

4.4. Model Kompleks

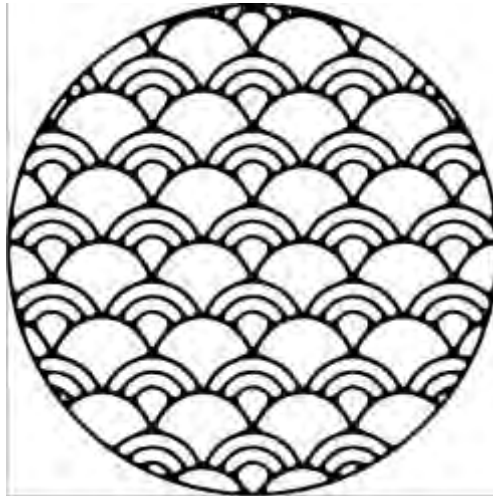
Model- model berikut dikategorikan dalam model sederhana karena waktu yang dibutuhkan oleh mesin CNC untuk mencetak model-model berikut relatif sangat lama.

4.4.1. Data Pertama

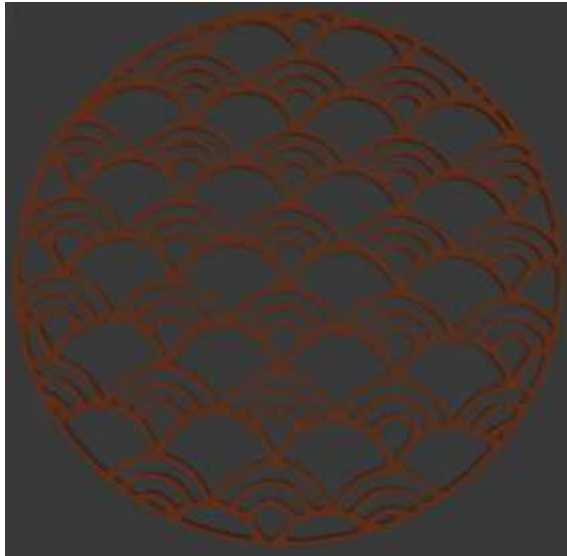
Data pertama dari kategori model kompleks merupakan suatu pola seperti pada gambar 4.25. Grafis komputer 3 dimensi seperti pada gambar 4.26 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.27.

Face dan *vertex* dari grafis komputer pada gambar 4.26 yang terdefinisikan pada *file* .obj berjumlah 26 buah *face* dan 10140 buah *vertex*.

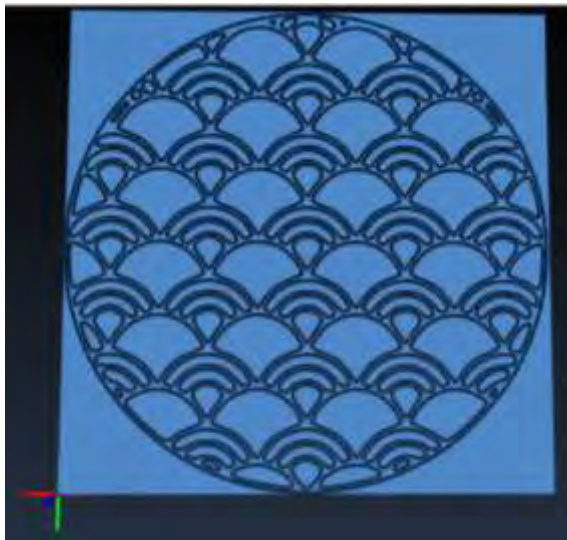
Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.28. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 38 menit dan 8 detik.



Gambar 4.29 Gambar 2 dimensi dari data pertama



Gambar 4.30 Model 3 dimensi dari data pertama



Gambar 4.31 Hasil simulasi dari data pertama



Gambar 4.32 Obyek yang dihasilkan mesin CNC

Karena ukuran yang terlalu kecil dan material yang rapuh, terdapat ketidaksesuaian antara obyek yang dihasilkan dengan grafik komputer 3 dimensi.

4.4.2. Data Kedua

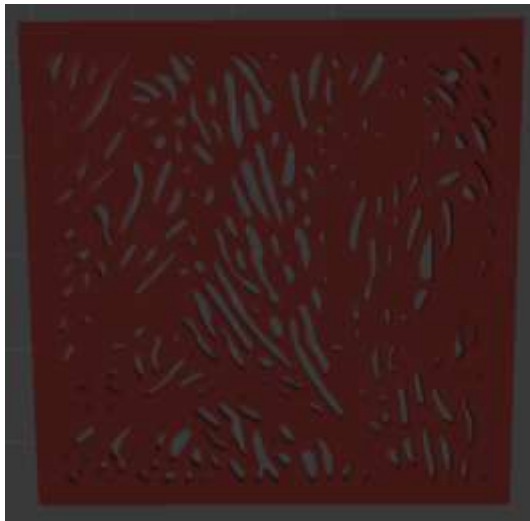
Data kedua dari kategori model kompleks merupakan suatu pola seperti pada gambar 4.29. Grafis komputer 3 dimensi seperti pada gambar 4.30 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.31.

Face dan *vertex* dari grafis komputer pada gambar 4.30 yang terdefinisikan pada *file* .obj berjumlah 100 buah *face* dan 10791 buah *vertex*.

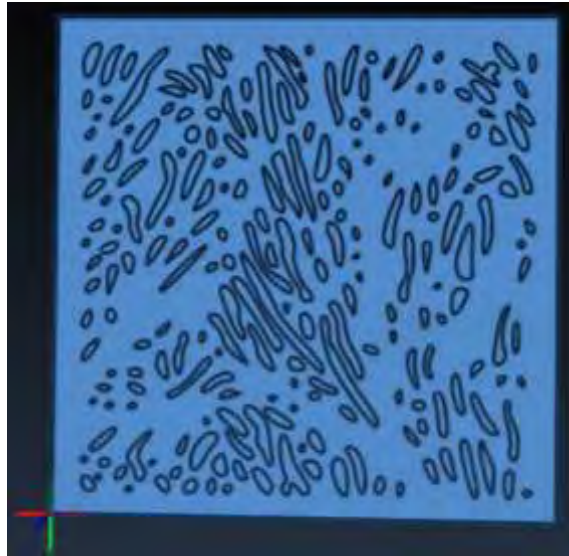
Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.32. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 69 menit dan 44 detik.



Gambar 4.33 Gambar 2 dimensi dari data kedua



Gambar 4.34 Gambar 4.35 Model 3 dimensi dari data kedua



Gambar 4.36 Hasil simulasi dari data kedua



Gambar 4.37 Obyek yang dihasilkan mesin CNC

4.5. Proses *Embossing*

Terdapat dua buah data yang diuji pada proses *embossing*, yaitu data kedua dan ketiga dari model sederhana pada proses *cutting*.

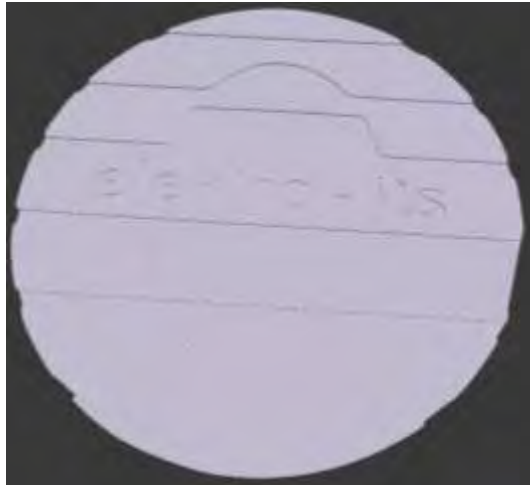
4.5.1. Data Pertama

Data pertama dari proses *embossing* merupakan logo dari Jurusan Teknik Elektro Institut Teknologi Sepuluh Nopember seperti pada gambar 4.37. Grafis komputer 3 dimensi seperti pada gambar 4.38 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file* .gcode yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.39.

Ketika *file* .gcode yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.40. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 16 menit dan 25 detik.



Gambar 4.38 Gambar 2 dimensi dari data pertama



Gambar 4.39 Model 3 dimensi dari data pertama



Gambar 4.40 Hasil simulasi dari data pertama



Gambar 4.41 Obyek yang dihasilkan mesin CNC

4.5.2. Data Kedua

Data kedua dari proses *embossing* merupakan logo dari Institut Teknologi Sepuluh Nopember seperti pada gambar 4.41. Grafis komputer 3 dimensi seperti pada gambar 4.42 diolah oleh perangkat lunak yang dirancang sehingga menghasilkan *file .gcode* yang ketika disimulasikan akan menghasilkan obyek seperti pada gambar 4.43.

Ketika *file .gcode* yang dihasilkan dimasukkan pada mesin CNC, dihasilkan obyek seperti pada gambar 4.44. Proses pencetakan mesin CNC untuk data pertama ini membutuhkan waktu 17 menit dan 16 detik.



Gambar 4.42 Gambar 2 dimensi dari data kedua



Gambar 4.43 Gambar 4.44 Model 3 dimensi dari data kedua



Gambar 4.45 Hasil simulasi dari data kedua



Gambar 4.46 Obyek yang dihasilkan mesin CNC

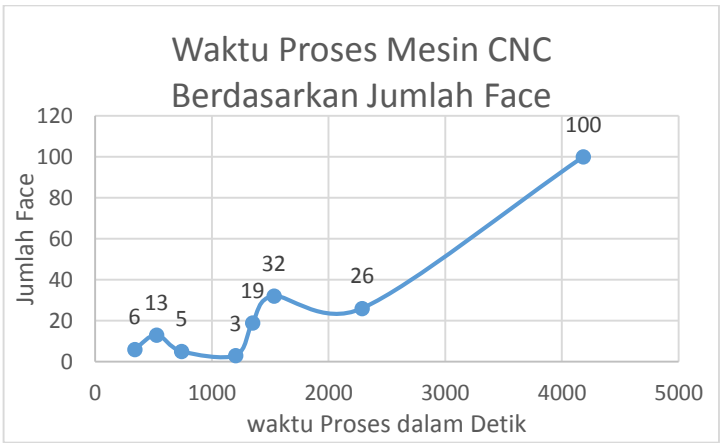
4.6. Analisa Data

4.6.1. Kuantitas

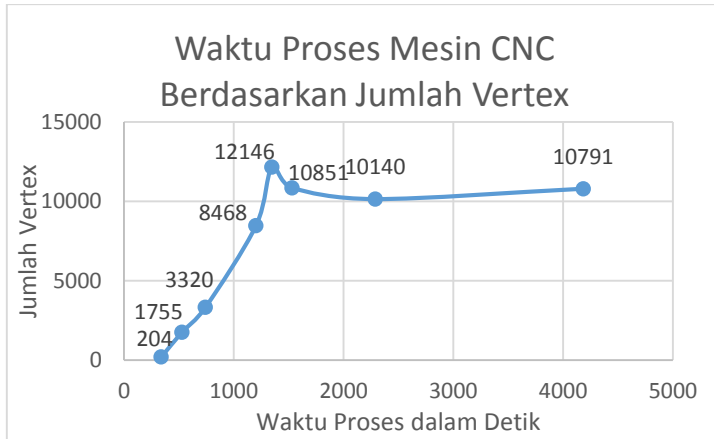
Berikut merupakan tabel dan grafik dari data yang diperoleh dari pengujian data terhadap 8 model 3 dimensi pada tiga kategori.

Model	Jumlah Face	Jumlah vertex	Waktu Proses
1	6	204	338
2	13	1755	526
3	5	3320	740
4	3	8468	1202
5	19	12146	1348
6	32	10851	1532
7	26	10140	2288
8	100	10791	4184

Tabel 4.2 *Face*, *vertex*, dan waktu proses dari ketiga kategori model



Gambar 4.47 Grafik hubungan jumlah *face* dengan waktu proses



Gambar 4.48 Grafik hubungan jumlah *vertex* dengan waktu proses

Berdasarkan data yang didapat, semakin banyak jumlah *face* dan *vertex* waktu proses mesin CNC rata – rata semakin besar. Waktu rata-rata proses tiap *face* adalah 0.015246 *face* tiap detik atau dibutuhkan rata-rata 65.59 detik. Sedangkan waktu rata-rata proses tiap *vertex* adalah 4.82196 *vertex* tiap detik.

4.6.2. Kualitas

Obyek yang dihasilkan mesin CNC sesuai dengan model 3 dimensi, namun jika dilihat lebih detail, obyek yang dihasilkan mesin CNC tidak sesuai dengan model 3 dimensi. Hal ini dapat dilihat pada gambar 4.48.



Gambar 4.49 Perbandingan hasil simulasi dan obyek cetak

Terdapat beberapa kemungkinan yang menyebabkan hal ini, antara lain *feedrate* yang terlalu tinggi, jumlah *vertex* yang terlalu padat pada satu tempat tertentu. dan presisi serta akurasi dari mesin.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

5.1. Kesimpulan

Dari pengujian dan hasil analisis tugas akhir ini dapat diambil beberapa kesimpulan untuk sistem yang dibangun.

1. Jalur pergerakan *tool* dari mesin CNC bergantung pada *face* dan *vertex* yang terdefinisikan pada *file* yang merepresentasikan grafis komputer 3 dimensi.
2. Dengan ukuran, karakteristik, dan jumlah *face* yang sama, semakin banyak jumlah *vertex*, waktu pemrosesan dari mesin CNC akan meningkat.
3. Dengan karakteristik dan jumlah *vertex* yang sama, semakin banyak jumlah *face*, waktu pemrosesan dari mesin CNC akan meningkat.
4. Waktu proses mesin CNC rata-rata ketiga kategori pada material berbahan *medium density fibreboard* atau MDF dengan luas 30 cm x 30 cm adalah 535 detik, 1361 detik, dan 3236 detik.
5. Hasil obyek cetak yang dihasilkan secara umum menyerupai model 3 dimensi, namun secara detail tidak sesuai.

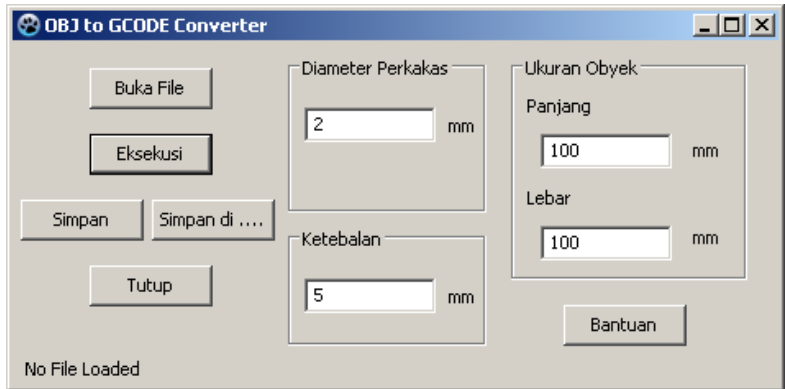
5.2. Saran

1. Agar waktu pencetakan benda lebih singkat, dapat dilakukan optimasi jalur pergerakan *tool*.
2. G-code yang digunakan pada perangkat lunak di Tugas Akhir ini terbatas sehingga perlu ditambahkan lebih banyak kode.
3. Perlu ditambahkan fitur untuk membaca *file* 3 dimensi dengan berbagai format lain.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN

1. Tampilan Perangkat Lunak



2. Kode

```
unit Unit1;

{$mode objfpc}{$H+}

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics,
  Dialogs, StdCtrls, Math;

type
  CustomArray      =      Array of Array[0..3] of Real;
  { TForm1 }

  TForm1 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    Button6: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    GroupBox1: TGroupBox;
```

```

GroupBox2: TGroupBox;
GroupBox3: TGroupBox;
Label1: TLabel;
Label2: TLabel;
Label4: TLabel;
Label5: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Memo2: TMemo;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Read;
procedure Sort(var ar : CustomArray);

procedure Create;
procedure Write;
procedure Button1Click(Sender: TObject);
private
{ private declarations }
public
{ public declarations }
end;

var
Form1: TForm1;
myfile      :      TextFile;
str,dir      :      String;
list1       :      TStringList;
v           :      Array of Array[0..3] of Real;
f,e,es,fs,ee :      Array of Array of Integer;
v1,vll      :      Array of Array of Array[0..3]
of Real;
d,b,t,w      :      Real;

implementation

{$R *.lfm}

Function Min(arr : CustomArray; k : Byte) : Real;
var i : integer; j : real;
begin
Result := 9999999;
for j := 1 to High(arr) do
if Result >= arr[i,k] then Result := arr[i,k];
end;

Function Max(arr : CustomArray; k : Byte) : Real;
var i : integer; j : real;
begin
Result := 0;

```

```

for i := 1 to High(arr) do
    if Result <= arr[i,k] then Result := arr[i,k];
end;

Function Check(a,aa : Integer; c : Array of Real) : Integer;
var i : integer; h : real;
begin
    Result := 0;

    for i := 1 to aa-1 do
        if ((v1[a,i,1] < c[0]) and (v1[a,i+1,1] > c[0])) then
            if (v1[a,i,2] > c[1]) and (v1[a,i+1,2] > c[1]) then
                Result := Result + 1
            else if ((v1[a,i,2] < c[1]) and (v1[a,i+1,2] > c[1]))
            or ((v1[a,i,2] > c[1]) and (v1[a,i+1,2] < c[1])) then
                begin
                    h := (v1[a,i+1,2] - v1[a,i,2])*(c[0] -
                    v1[a,i,1])/(v1[a,i+1,1] - v1[a,i,1]) + v1[a,i,2];
                    if h > c[1] then Result := Result + 1;
                end;

            if ((v1[a,aa,1] < c[0]) and (v1[a,1,1] > c[0])) then
                if (v1[a,aa,2] > c[1]) and (v1[a,1,2] > c[1]) then
                    Result := Result + 1
                else if ((v1[a,aa,2] < c[1]) and (v1[a,1,2] > c[1]))
                or ((v1[a,aa,2] > c[1]) and (v1[a,1,2] < c[1])) then
                    begin
                        h := (v1[a,1,2] - v1[a,aa,2])*(c[0] -
                        v1[a,aa,1])/(v1[a,1,1] - v1[a,aa,1]) + v1[a,aa,2];
                        if h > c[1] then Result := Result + 1;
                    end;

            for i := aa downto 1 do
                if ((v1[a,i,1] < c[0]) and (v1[a,i-1,1] > c[0])) then
                    if (v1[a,i,2] > c[1]) and (v1[a,i-1,2] > c[1]) then
                        Result := Result + 1
                    else if ((v1[a,i,2] < c[1]) and (v1[a,i-1,2] > c[1]))
                    or ((v1[a,i,2] > c[1]) and (v1[a,i-1,2] < c[1])) then
                        begin
                            h := (v1[a,i-1,2] - v1[a,i,2])*(c[0] -
                            v1[a,i,1])/(v1[a,i-1,1] - v1[a,i,1]) + v1[a,i,2];
                            if h > c[1] then Result := Result + 1;
                        end;

            end;
        { TForm1 }

        procedure TForm1.Read;
        var h,i,j,k : integer; l : array of integer; m,n : real;
        begin

            list1 := TStringList.Create;
            list1.Delimiter := ',';
            assignfile(myfile, OpenFileDialog1.FileName);

```

```

reset(myfile);
i := 0;
h := 0;
k := 0;

while not eof(myfile) do
begin
  readln(myfile, str);
  list1.DelimitedText := str;
  if list1[0] = 'v' then
  begin
    i := i + 1;
    SetLength(v, i+1);
    for j:=1 to 3 do v[i,j] := strttoFloat(list1[j]);
    v[i,0] := i;
  end;
end;
closeFile(myfile);

j := 0;
for i := 1 to High(v) do
  if v[i,3] = Min(v,3) then h := h + 1
  else if v[i,3] = Max(v,3) then j := j + 1
  else if (v[i,3] < Max(v,3)) and (v[i,3] > Min(v,3)) then
    n := v[i,3];

if j <= h then m := Max(v,3) else m := Min(v,3);

assignfile(myfile, OpenFileDialog1.FileName);
reset(myfile);
h := 0;
SetLength(fs, 1);
SetLength(f, 1);
while not eof(myfile) do
begin
  readln(myfile, str);
  list1.DelimitedText := str;
  if list1[0] = 'f' then
  begin
    SetLength(f, Length(f) + 1);
    SetLength(f[High(f)], list1.count);
    for j := 1 to list1.count-1 do f[High(f),j] :=
      strtoint(list1[j]);

    i := 0;
    for j := 1 to High(f[High(f)]) do
      if v[f[High(f),j],3] <> n then
      begin
        i := i + 1;
      end;

    if i = 0 then
    begin
      SetLength(fs, Length(fs) + 1);

```

```

    fs[High(fs)] := f[High(f)];
end;

for j := 1 to High(f[High(f)]) do
    if (v[f[High(f),j],3] <= m) then
        begin
            SetLength(f, Length(f) - 1);

            break;
        end;
    end;
end;
closeFile(myfile);

SetLength(e, Length(v));
SetLength(es, Length(v));
for i := 1 to High(e) do
    begin
        SetLength(e[i],1);
        SetLength(es[i],1);
    end;

for i := 1 to High(f) do
    begin
        for j := 2 to High(f[i])-1 do
            begin
                SetLength(e[f[i,j]], Length(e[f[i,j]])+1);
                e[f[i,j],High(e[f[i,j]])] := f[i,j+1];
                SetLength(e[f[i,j]], Length(e[f[i,j]])+1);
                e[f[i,j],High(e[f[i,j]])] := f[i,j-1];
            end;
            SetLength(e[f[i,1]], Length(e[f[i,1]])+1);
            e[f[i,1],High(e[f[i,1]])] := f[i,2];

            SetLength(e[f[i,1]], Length(e[f[i,1]])+1);
            e[f[i,1],High(e[f[i,1]])] := f[i,High(f[i])];

            SetLength(e[f[i,High(f[i])]],
                Length(e[f[i,High(f[i])]])+1);
            e[f[i,High(f[i])],High(e[f[i,High(f[i])])]] := f[i,1];

            SetLength(e[f[i,High(f[i])]],
                Length(e[f[i,High(f[i])]])+1);
            e[f[i,High(f[i])],High(e[f[i,High(f[i])])]] :=
                f[i,High(f[i])-1];
        end;

for h := 1 to High(e) do
    begin
        for i := 1 to High(e[h])-1 do
            begin
                for j := i to High (e[h])-1 do

```

```

        if e[h,i] = e[h,j+1] then
            begin
                SetLength(es[h], Length(es[h])+1);
                es[h,High(es[h])] := e[h,i];

                for k := j+1 to High(e[h])-1 do
                    begin
                        e[h,k] := e[h,k+1];
                    end;
                SetLength(e[h], Length(e[h])-1);

            end;
        end;
    end;
end;

end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    SetLength(v, 0);
    SetLength(f, 0);
    SetLength(e, 0);
    SetLength(es, 0);
    SetLength(vl, 0);

    if OpenFileDialog1.Execute then
        begin
            Read;
            ShowMessage('File telah diambil dari file '+
                OpenFileDialog1.FileName);
            dir := OpenFileDialog1.InitialDir;
            Label9.Caption := OpenFileDialog1.FileName+' Loaded'
        end;

end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    Close;
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
    Showmessage('Tombol "Buka File" untuk membuka file obj' +
        #13#10 +
        'Tombol "Eksekusi" untuk memulai proses konversi' + #13#10
        +
        'Tombol "Simpan" untuk menyimpan file G-code pada
        directory file obj dengan nama yang sama' + #13#10 +
        'Tombol "Simpan di ...." untuk menyimpan file G-Code' +
        #13#10 + #13#10 +
        'Masukkan diameter perkakas atau mata bor pada "Diameter
        Perkakas"' + #13#10 +

```

```

'Masukkan kedalam obyek cetak yang diinginkan pada
  "Ketebalan"' + #13#10 +
'Masukkan dimensi material cetak pada "Ukuran Obyek"' +
  #13#10 +
'Status proses dapat dilihat pada bagian bawah kiri dari
  program'+ #13#10 + #13#10 +
'Jurusan Teknik Elektro'+ #13#10 +
'Fakultas Teknologi Industri'+ #13#10 +
'Institut Teknologi Sepuluh Nopember'+ #13#10 +
'Suraaya'+ #13#10 +
'2014');
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
  ChDir(dir);
  Reset(myfile);
  list1 := TStringList.Create;
  list1.Delimiter := '\';
  list1.DelimitedText := OpenFileDialog1.FileName;
  str := list1[list1.Count-1];
  list1.Delimiter := '.';
  list1.DelimitedText := str;
  str := list1[0]+'.gcode';
  AssignFile(myfile, str);
  Write;
  Label9.Caption := 'File saved at ' + dir
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
  saveDialog1.Filter := 'G-code file|*.gcode|MKverge GUI
    file|*.mmg';

  saveDialog1.DefaultExt := 'gcode';
  if SaveDialog1.Execute then
  begin
    Reset(myfile);
    AssignFile(myfile, SaveDialog1.FileName);
    Write;
    ShowMessage('Data telah disimpan pada file ' +
      SaveDialog1.FileName);
    Label9.Caption := 'File saved at ' +
      Savedialog1.InitialDir;
  end
  else
    Label9.Caption := 'No File Saved';
  end;

procedure TForm1.Sort(var ar : CustomArray);
var i,k : integer; l : array[0..3] of real; j,m,n : real;
begin
  n := Min(ar,1);
  j := Min(ar,2);
  m := Min(ar,3);

```



```

for i := High(ar) downto 1 do
begin
ar[i,1] := ar[i,1] - n;
ar[i,2] := ar[i,2] - j;
ar[i,3] := ar[i,3] - m;
end;
n := max(ar, 1);
if strtfloat(Edit1.Text)/Max(ar,1) <=
  strtfloat(Edit2.Text)/Max(ar,2) then
  j := (strtfloat(Edit1.Text)-2*t)/Max(ar,1)
else
  j := (strtfloat(Edit2.Text)-2*t)/Max(ar,2);

d := - strtfloat(Edit5.Text);
for i := 1 to High(ar) do
begin
for k := 1 to 2 do ar[i,k] := ar[i,k]*j+t;
if (Max(ar,3) = 0) or (Max(ar,3) = d) then ar[i,3] := d
else ar[i,3] := d;
end;

end;

procedure TForm1.Create;
var g,h,i,j,k : integer; l : array of array[0..3] of real;
w,r,n,o,n1,o1,n2,o2 : real;
m : array[1..2] of array[1..2] of real; ee1,ee2 :
  Array of Array of Integer;
te : array of integer;
  label y,yy,yyy;
begin
d := - strtfloat(Edit5.Text);
r := 3;

SetLength(vl, Length(vl)+1);
for i := 1 to High(f) do
begin
SetLength(vl, Length(vl) + 1);
SetLength(vl[i], 1);

for j := 1 to High(f[i])-1 do
begin

SetLength(vl[i], Length(vl[i])+1);
vl[i,High(vl[i])] := v[f[i,j]];

SetLength(vl[i], Length(vl[i])+1);
vl[i,High(vl[i])] := v[f[i,j+1]];

for k := 1 to High(es[f[i,j]]) do
begin
if es[f[i,j],k] = f[i,j+1] then
begin
vl[i,High(vl[i])-1,3] := r;

```

```

        vl[i,High(vl[i]),3] := r;
        break;
    end;
end;

end;

SetLength(vl[i], Length(vl[i])+1);
vl[i,High(vl[i])] := v[f[i,High(f[i])]];

SetLength(vl[i], Length(vl[i])+1);
vl[i,High(vl[i])] := v[f[i,1]];

for k := 1 to High(es[f[i,High(f[i])]]) do
begin
    if es[f[i,High(f[i])],k] = f[i,1] then
    begin
        vl[i,High(vl[i])-1,3] := r;
        vl[i,High(vl[i]),3] := r;
    end;
end;

end;

for i := 1 to High(vl) do
for j := 1 to High(vl[i])-1 do
    if (vl[i,j,0] = vl[i,j+1,0]) and (vl[i,j,3] =
        vl[i,j+1,3]) then
    begin
        if (vl[i,j,3] <= 0.0000001) and (vl[i,j,3] >= -
            0.0000001) then break;
        for g := j+1 to High(vl[i]) do
            vl[i,g] := vl[i,g+1];
        SetLength(vl[i], Length(vl[i]) - 1);
    end;
SetLength(vl[i], 1);
k := High(vl);
for i := 1 to High(fs) do
begin
    SetLength(vl[i], Length(vl[i]) + 1);
    SetLength(vl[i][High(vl[i])], Length(vl[i][High(vl[i])]) + 1);
    for j := 1 to High(fs[i]) do
    begin
        SetLength(vl[i][High(vl[i])], Length(vl[i][High(vl[i])]) +
            1);
        vl[i][High(vl[i]),j] := v[fs[i,j]];
    end;
end;
SetLength(ee, Length(v));
for i := 1 to High(ee) do
begin
    SetLength(ee[i],1);
end;

for i := 1 to High(fs) do

```

```

begin
for j := 2 to High(fs[i])-1 do
begin
SetLength(ee[fs[i,j]], Length(ee[fs[i,j]])+1);
ee[fs[i,j],High(ee[fs[i,j]])] := fs[i,j]+1;
SetLength(ee[fs[i,j]], Length(ee[fs[i,j]])+1);
ee[fs[i,j],High(ee[fs[i,j]])] := fs[i,j]-1;
end;
SetLength(ee[fs[i,1]], Length(ee[fs[i,1]])+1);
ee[fs[i,1],High(ee[fs[i,1]])] := fs[i,2];

SetLength(ee[fs[i,1]], Length(ee[fs[i,1]])+1);
ee[fs[i,1],High(ee[fs[i,1]])] := fs[i,High(fs[i])];

SetLength(ee[fs[i,High(fs[i])]],
Length(ee[fs[i,High(fs[i])]])+1);
ee[fs[i,High(fs[i])],High(ee[fs[i,High(fs[i])]])] :=
fs[i,1];

SetLength(ee[fs[i,High(fs[i])]],
Length(ee[fs[i,High(fs[i])]])+1);
ee[fs[i,High(fs[i])],High(ee[fs[i,High(fs[i])]])] :=
fs[i,High(fs[i])-1];
end;
for i := 1 to High(fs) do
begin
SetLength(v1, Length(v1)+1);
SetLength(v1[High(v1)], 1);
n1 := Min(v1[i],1);n2 := Max(v1[i],1);
o1 := Min(v1[i],2);o2 := Max(v1[i],2);

w := n1-t;
repeat
w := w + t;
SetLength(ee1, 0);
SetLength(ee1,1);
SetLength(ee2, 0);
SetLength(ee2,1);
for j := 1 to High(ee) do
begin
if (Length(ee) > 1) and (ee[j,1] > 0) and (ee[j,1] <
Length(ee)) then
if ((w >= v[j,1]) and (w <= v[ee[j,1],1])) or ((w <=
v[j,1]) and (w >= v[ee[j,1],1])) then
begin
SetLength(ee1, Length(ee1) + 1);
SetLength(ee1[High(ee1)], 3);
ee1[High(ee1),1] := j;
ee1[High(ee1),2] := ee[j,1];
end;
end;
g := 0; SetLength(ee2, 1);
for j := 1 to High(ee1) do
begin
for h := 1 to High(fs[i]) do

```

```

        if (ee1[j,1] = fs[i,h]) or (ee1[j,2] = fs[i,h])
then g := g + 1;

if g > 0 then
begin
    SetLength(ee2, Length(ee2)+1);
    ee2[High(ee2)] := ee1[j];
end;
end;
repeat
j := 0;
for g := 1 to High(ee2)-1 do
begin
    if v[ee2[g,1],2] > v[ee2[g,2],2] then n :=
v[ee2[g,1],2] else n := v[ee2[g,2],2];

    if v[ee2[g+1,1],2] > v[ee2[g+1,2],2] then o :=
v[ee2[g+1,1],2] else o := v[ee2[g+1,2],2];

    if n < o then
begin
        te := ee2[g];
        ee2[g] := ee2[g+1];
        ee2[g+1] := te;
        j := j + 1;
end;

end;
until j = 0;

j := 1;
o := ((w - v[ee2[j,2],1])*(v[ee2[j,1],2]-
v[ee2[j,2],2]))/(v[ee2[j,1],1]-v[ee2[j,2],1])) +
v[ee2[j,2],2];
SetLength(v1[High(v1)], Length(v1[High(v1)]) + 1);
v1[High(v1),High(v1[High(v1)]),1] := w;
v1[High(v1),High(v1[High(v1)]),2] := o;
v1[High(v1),High(v1[High(v1)]),3] := r;
n := d;
for j := 1 to High(ee2) do
begin
    o := ((w - v[ee2[j,2],1])*(v[ee2[j,1],2]-
v[ee2[j,2],2]))/(v[ee2[j,1],1]-v[ee2[j,2],1])) +
v[ee2[j,2],2];

SetLength(v1[High(v1)], Length(v1[High(v1)]) + 1);
v1[High(v1),High(v1[High(v1)]),1] := w;
v1[High(v1),High(v1[High(v1)]),2] := o;
v1[High(v1),High(v1[High(v1)]),3] := n;

    if j mod 2 = 0 then
begin
        n := r;
        o := ((w - v[ee2[j,2],1])*(v[ee2[j,1],2]-
v[ee2[j,2],2]))/(v[ee2[j,1],1]-v[ee2[j,2],1])) +
v[ee2[j,2],2];

```

```

1);      SetLength(v1[High(v1)], Length(v1[High(v1)]) +
        v1[High(v1),High(v1[High(v1)])],1] := w;
        v1[High(v1),High(v1[High(v1)])],2] := o;
        v1[High(v1),High(v1[High(v1)])],3] := n;
    end
    else if (j mod 2 = 1) and (j > 1) then
    begin
        n := d;
        o := ((w - v[ee2[j,2],1])*(v[ee2[j,1],2]-
v[ee2[j,2],2])/(v[ee2[j,1],1]-v[ee2[j,2],1])) +
v[ee2[j,2],2];
        SetLength(v1[High(v1)], Length(v1[High(v1)]) +
1);
        v1[High(v1),High(v1[High(v1)])],1] := w;
        v1[High(v1),High(v1[High(v1)])],2] := o;
        v1[High(v1),High(v1[High(v1)])],3] := n;
    end;

end;
j := High(ee2);
m[1,2] := (w - v[ee2[j,1],1])*(v[ee2[j,2],2]-
v[ee2[j,1],2])/(v[ee2[j,2],1]-v[ee2[j,1],1])+
v[ee2[j,1],2];
SetLength(v1[High(v1)], Length(v1[High(v1)]) + 1);
v1[High(v1),High(v1[High(v1)])],1] := w;
v1[High(v1),High(v1[High(v1)])],2] := o;
v1[High(v1),High(v1[High(v1)])],3] := r;
//end;

until w >= n2-t;

end;

for i := 1 to High(v1) do
begin
k := 0;
for j := 1 to High(v1[i]) do
    if v1[i,j,0] = 0 then k := k + 1;
    if k div 2 = 1 then
    begin
        for k := i to High(v1)-1 do
        begin
            v1[i] := v1[k+1];
        end;
        SetLength(v1, Length(v1) - 1);
    end;
end;
end;

procedure TForm1.Write;
var i,j,k : integer; f,r,s : real; g : double;

```

```

begin

f := 1000;
r := 3;
s := 1000;

Rewrite(myfile);
writeln(myfile, '%');
writeln(myfile, 'G00'+x0'+y0'+z'+floattostr(r));
writeln(myfile, 'M03');

for i := 1 to High(v1) do
begin
f := 5000;
writeln(myfile,
'G00'+x'+floattostr(v1[i,1,1])+y'+floattostr(v1[i,1,2
])+z'+floattostr(r));

for j := 1 to High(v1[i]) do
begin
if v1[i,j,3] = r then f := 1500 else f := 1200;
if (j = 1) and (v1[i,j,3] <> r) then f := 1500;
if (j > 1) and (v1[i,j-1,3] = r) and (v1[i,j,3] <>
r) then f := 1500;
writeln(myfile, 'G01'+x'+floattostr(v1[i,j,1])+
y'+floattostr(v1[i,j,2])+z'+floattostr(v1[i,j,3])+
F'+floattostr(f));

end;

writeln(myfile, 'G00'+z'+floattostr(r));
end;

for i := 1 to High(v11) do
begin
f := 5000;
writeln(myfile,
'G00'+x'+floattostr(v11[i,1,1])+y'+floattostr(v11[i,1
,2])+z'+floattostr(r));

for j := 1 to High(v11[i]) do
begin
if v11[i,j,3] = r then f := 1500 else f := 1200;
if (j = 1) and (v11[i,j,3] <> r) then f := 1500;
if (j > 1) and (v11[i,j-1,3] = r) and (v11[i,j,3] <>
r) then f := 1500;
writeln(myfile, 'G01'+x'+floattostr(v11[i,j,1])+
y'+floattostr(v11[i,j,2])+z'+
'+floattostr(v11[i,j,3])+F'+floattostr(f));

end;

```

```

    writeln(myfile, 'G00'+ 'Z'+floattostr(r));
  end;
writeln(myfile, '%');
closeFile(myfile);

end;

procedure TForm1.Button1Click(Sender: TObject);
var i,j : integer;
begin
  Label9.Caption := 'Processing...';
  t := strtofloat(Edit4.Text);
  b := t/2;
  d := - strtofloat(Edit5.Text);

  Sort(v);
  Create;

  Label9.Caption := 'Done !';

end;

end.

```

DAFTAR PUSTAKA

1. Tobler & Maierhofer, "A Mesh Data Structure for Rendering and Subdivision", 2006
2. _____, "*Object Files (.obj)*" <URL : <http://www.martinreddy.net/gfx/3d/OBJ.spec>>, November, 2007
3. Smid, Peter, "*CNC Programming Handbook*", New York: Industrial Press, 2008
4. _____, "*GCODE Introudction for Programmers*" <URL : <http://cncutil.org/gcode-introduction.html>>, November, 2007
5. _____, "*Blender (Software)*" <URL : http://en.wikipedia.org/wiki/Blender_%28software%29>, Januari, 2015
6. _____, "*OpenSCAM*" <URL : <http://openscam.com/>>, November, 2007
7. _____, "*Lazarus*" <URL : http://en.wikipedia.org/wiki/Lazarus_%28IDE%29>, Januari, 2015
8. _____, "*Mastercam*" <URL : <http://en.wikipedia.org/wiki/Mastercam>>, Desember, 2014
9. _____, "*Excitech SHG1224 Product Specification*" <URL : <http://www.excitech.cn/products.aspx?id=37>>, November, 2007

-----Halaman ini sengaja dikosongkan-----

BIODATA PENULIS



Penulis bernama lengkap Stefan Feliciano Tandawidjaja lahir pada tanggal 15 September 1993, sebagai anak pertama dari dua bersaudara. Penulis mengawali kegiatan pendidikan formal di SDK Santa Maria Surabaya, yang kemudian dilanjutkan di SMPK Santa Maria Surabaya, kemudian SMAK St. Louis 1 Surabaya dan pada tahun 2011 penulis diterima sebagai mahasiswa di jurusan Teknik Elektro ITS. Saat kuliah, penulis turut berpartisipasi sebagai asisten praktikum di bidang studi elektronika.

Email : stefan.tandawidjaja@gmail.com

-----Halaman ini sengaja dikosongkan-----